The Effect of Interference on Delta Modulation Encoded Video Signals

Semi- Annual Report

October 1, 1978 - April 1, 1979

Goddard Space Flight Center

Greenbelt, Maryland

under

NASA Grant: NSG - 5013

Donald L. Schilling

Professor of Electrical Engineering

Principal Investigator

COMMUNICATIONS SYSTEMS LABORATORY

DEPARTMENT OF ELECTRICAL ENGINEERING

(NASA-CR-157960) THE EFFECT OF INTERFERENCE ON DELTA MODULATION ENCODED VIDEO SIGNALS Semiannual Report, 1 Oct. 1978 - 1 Apr. 1979 (City Coll. of the City Univ. of New York.) 161 p HC A08/MF A01 CSCI 17B G3/32

N79-18157

Unclas 16088

THE CITY COLLEGE OF
THE CITY UNIVERSITY of NEW YORK

The Effect of Interference on Delta Modulation Encoded Video Signals

Semi-Annual Report

October 1, 1978 - April 1, 1979

Goddard Space Flight Center

Greenbelt, Maryland

under

NASA Grant: NSG-5013

Donald L. Schilling

Professor of Electrical Engineering

Principal Investigator

COMMUNICATIONS SYSTEMS LABORATORY

DEPARTMENT OF ELECTRICAL ENGINERING

Table of Contents

INTRODUCTION

This report summarizes a portion of the research performed under NASA Grant NSG 5013 from October 1, 1978 - April 1, 1979. The report focuses on the response of several demodulators used to encode video. The report considers the effect of bit rate and error rate as well as the algorithm on the picture response.

The following doctoral students M. Braff, S. Davidovici, R. Dhadasoogar, and N. Scheinberg. The report was prepared by N. Scheinberg and represents his Doctoral Dissertation submitted to the Faculty of the City College of New York in January 1979.

Following a long history of cooperation between NASA and Dr. Schilling and his graduate students, a DM system has been constructed for NASA - Goddard. Using this system will enable NASA and Dr. Schilling to determine accurately the effects of interference on video signals which have been encoded using delta modulation. The system has the capability of encoding (and decoding) voice as well as video information. The voice being transmitted on the horizontal synch pulse.

Abstract

This paper presents the results of a study on the use of the delta modulator as a digital encoder of television signals. The study began with the computer simulation of different delta modulators in order to find a satisfactory delta modulator. After finding a suitable delta modulator algorithm via computer simulation, we analyzed the results, and then implemented it in hardware to study its ability to encode real time motion pictures from an NTSC format television camera.

We then investigated the effects of channel errors on the delta modulated video signal and tested several error correction algorithms via computer simulation. A very high speed delta modulator was built (out of ECL logic), incorporating the most promising of the correction schemes, so that it could be tested on real time motion pictures.

The final area of investigation concerned itself with finding delta modulators which could achieve significant bandwidth reduction without regard to complexity or speed. The first such scheme to be investigated was a real time "frame to frame" encoding scheme which required the assembly of fourteen, 131,000 bit long shift registers as well as a high speed delta modulator. The other schemes involved

two dimensional delta modulator algorithms.

I. THE DELTA MODULATION OF VIDEO SIGNALS

by

NORMAN SCHEINBERG

A dissertation submitted to the Graduate Faculty in Engineering in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York.

1979

CONTENTS

CHAPTER I

Introduction

This study is concerned with the development of a practical and cost effective digital alternative to existing analog television transmission systems. At the present time television signals are sent long distances in an analog fashion using vestigial side band AM or by using wide band FM. AM is used in both commercial television broadcasts and in cable transmission to our homes. FM, on the other hand, is used between ground stations and communication satellites as well as on all of NASA's manned space flight missions, including the space shuttle. Wide band FM is used for satellite and space TV transmission because less power is needed with FM as compared to AM, and power is at a premium on a transmitter placed in space.

If the analog FM system could be replaced by a digital system, several advantages would result:

I. There is an improvement in signal-to-noise ratio.

II. Time-division multiplexing becomes simple.

III. Digital signals can be transmitted over long distances through many repeater stations with less degradation than analog signals.

IV. Scrambling for secrecy becomes straightforward.

The improvement in signal-to-noise ratio obtained by switching from an FM system to a digital one will, of course, depend on the type of digital system used and on the criterion used to calculate the

signal to noise ratio. If, for the sake of convenience, we assume the digital system is PCM, and the RMS value of signal to noise ratio is used, then it can be shown that about a 7 db improvement will result from switching from FM to PCM. This result is derived in Appendix A.

It was stated earlier that digital systems can easily be time division multiplexed making it possible for one satellite to handle several users at one time. Normally, this advantage could be partially offset by the ability to frequency division multiplex analog signals. Some of our communication satellites, however, hard limit the incoming signal and thus produce interference between frequency division multiplexed channels. This fact, plus the savings in transmitter power, would apparently make PCM the system of choice over FM.

Unfortunately PCM has a major drawback that has made it unusable everywhere for standard, real time (30 frames/sec) TV transmission. The problem lies in the fact that PCM expands the 4 MHz, black and white, base band, TV bandwidth by a factor of 6, to 24 MHz. This is equivalent to a 48 mega-bit per second digital system. Very few transmission systems can operate at 48 mega-bits per second or have 48 MHz of bandwidth to dedicate to a single TV channel. With an FM system the bandwidth can be reduced at the expense of transmitter power, but with PCM the user must allot the required bandwidth (48 MHz) regardless of transmitter power. Therefore, most users either do not send video information or use FM, and either increase transmitter power or accept reduced quality pictures.

CHAPTER II

Bandwidth Reduction Methods

To overcome the large bandwidth requirement of PCM, researchers have used the statistical properties of pictures, as well as the psycho-visual properties of the eye, to reduce the bandwidth requirement of digital encoders.

2.1 Statistical Coding

Straight forward statistical encoding of a picture to reduce bandwidth would proceed as follows [1]. Assume the picture consists of a 512 by 512 array of picture elements which are quantized to 64 different brightness levels. Then the total number of possible pictures than can be represented is:

$$(64)^{(512)^2} = 2^{1,572,864} \approx 10^{500,000}$$

Most of the $10^{500,000}$ possible pictures have a very low probability of being sent. In fact, most of the pictures would not even be recognizable to a human observer but would appear as random dots. This is because real pictoral data contains significant structure, and structure is a departure from randomness.

Now conceive of an encoder that has stored a unique code word for each of the $10^{500,000}$ pictures. If we assign short code words to the more likely pictures and long code words to the least likely pictures,

we can make the average number of bits required per picture close to the entropy

$$\sum_{i=1}^{10^{500,000}} p_i \log p_i$$

where p_i is the probability of the ith picture. Schriber [2] suggests that the entropy of pictures is about 1 bit per pel.

Using pure statistical encoding, we can reduce the bandwidth of digital TV from 48 MHz for PCM to 8 MHz for the statistical encoder. Unfortunately such an encoder cannot be built since it would require $10^{500,000}$ words of memory. Even if each picture were broken up into subpictures of only 4 x 4, the statistical encoding of a subpicture would still require 10^{28} words of memory.

2.2 Psychovisual Coding

Psychovisual Coding acheives bandwidth reduction by removing that information from a picture that will not be missed by a human observer. By removing information from a picture, the picture is changed, i.e. distorted, but some types of distortion cannot be perceived by a human observer.

Schriber [2] and others have done extensive studies of the human visual system and have drawn the following conclusions.

1. Many pictures can be adequately represented with 64 brightness levels, i.e. 6 bits per picture element (pel).
2. The shape of an object is more important than its brightness.

Thus, it is more important to represent accurately the width of the stripes on a shirt than to convey their true shade of gray.

3. The frequency response of the eye falls off at both low and high frequencies. Hence, the number of brightness levels required for fine detail is less than that for moderate detail.
4. The eye cannot perceive fine detail on a moving object and it takes about .6 seconds for the eye to adjust to a complete scene change (3).

2.3 Transform Coding

Transform coding is a method for accomplishing some aspects of both statistical and psychovisual coding. The first step in transform coding requires that the picture be sampled and digitized to at least 6 bits. The samples, (pels), are then arranged in a matrix such that the location of the pel in the picture corresponds to its location in the matrix. We then transform the pictures, represented as an n x n matrix, [X], into a new n x n matrix, [Y]. If the transformation is chosen properly, some of the elements of [Y] can be discarded and others quantized coarsely. The resulting [Y]* matrix could then be transmitted to the receiver using fewer bits than would have been necessary if [X], the original picture was sent. At the receiver the [Y]* matrix is inverse transformed back into a picture [X]*. If the transform was chosen properly, and if no important elements of [Y] were discarded,

then [X]*, the received picture, should look nearly the same as [X], the original picture.

To illustrate how transform coding works on pictures, we will begin with the familiar one dimensional discrete Fourier transform (DFT). The basis vectors for this transform are harmonically related sinusoids. To perform the transform we correlate our signal, x(n), n = 0, 1, 2... N-1, with each sinusoid. The result of each correlation is a number, or coefficient, Y(k). Mathematically we may write the correlation as:

$$Y(k) = \frac{1}{N} \sum_{n=0}^{N-1} e^{-J2\pi kn/N} x(n) \qquad (2.3.1)$$

$$k = 0, 1, 2, \ldots N-1$$

The exponential term is the sinusoid, k is the frequency, and x(o), x(1)...x(n-1) is the amplitude of the first, second and last sample of the signal to be transformed.

We may rewrite equation 2.3.1 in matrix notation as:

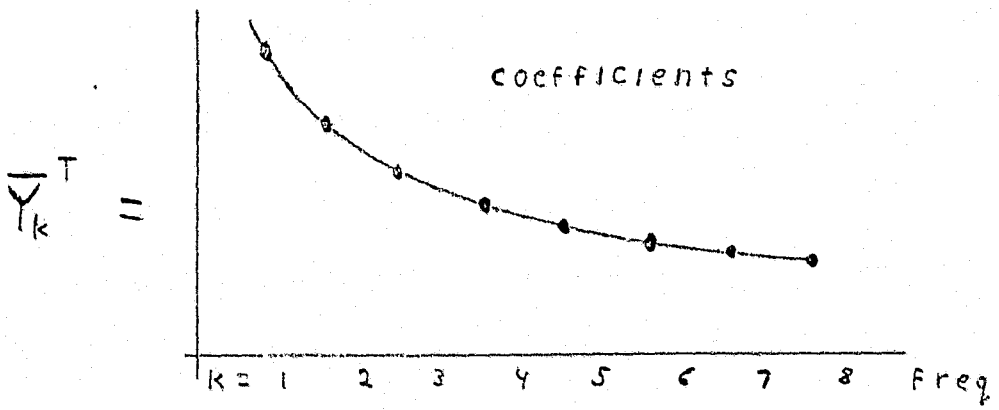
$$\overline{Y} = \frac{1}{N} [F] \overline{X} \qquad (2.3.2)$$

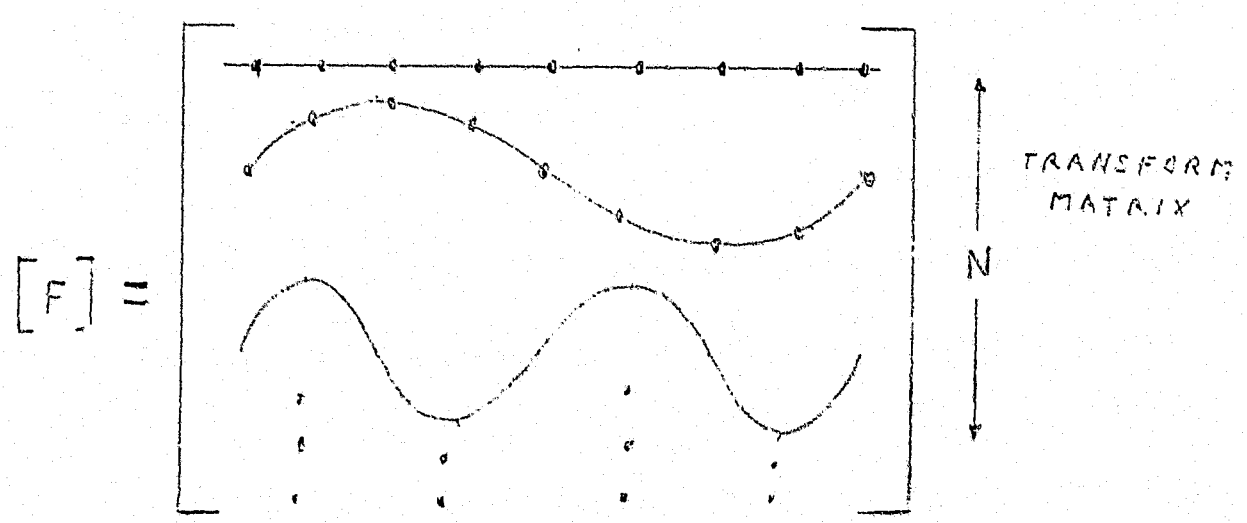
where $\overline{X}$ is a vector whose elements could be the brightness levels of the pels in a line of a picture. The vector, $\overline{Y}$, is the transform of $\overline{X}$, and [F] is an N x N matrix whose elements are given by 2.3.3

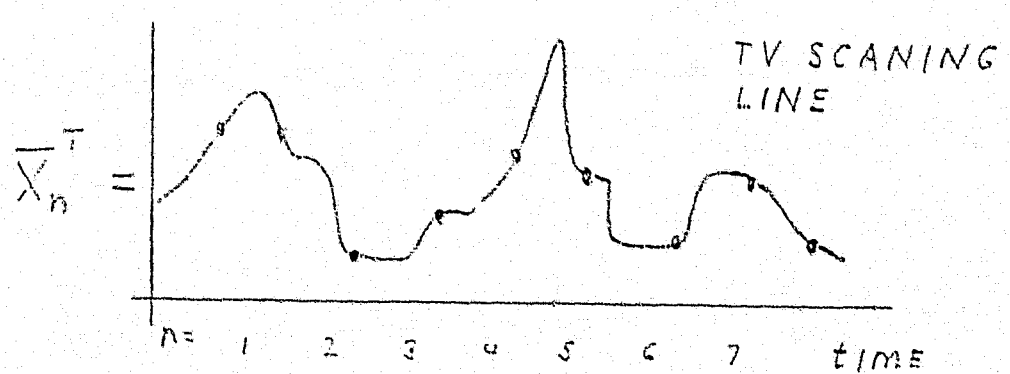
$$F_{ij} = e^{-j2\pi ij/N} \qquad i, j = 0, 1, 2, \ldots N-1 \qquad (2.3.3)$$

The amplitude of the elements of equation 2.3.2 are pictured in Figure I.

Fig. 1 Transform Coding







$\overline{X}$ may be recovered from $\overline{Y}$ by simply multiplying each coefficient of Y, y(k), k=0, 1, 2 ...N-1, by its corresponding sinusoid (basis vector) and summing up the result. In matrix notation we would write it as

$$[F]^T \overline{Y} = \overline{X} \tag{2.34}$$

(note $[F]^T = [F]^{-1}$ for this matrix)

At this point we have not saved any bandwidth since there are as many coefficients in the transform, $\overline{Y}$, as there were samples in the original line, $\overline{X}$. Bandwidth can be saved, however, if we discard some of the coefficients of $\overline{Y}$ and quantize others coarsely. It turns out that the statistics of pictures are such, that many of the coefficients are very small. Since these coefficients are small many of them can be discarded with little effect on the picture. From the psychophysics of vision we find that the high frequency components can be quantized coarsely since the eye is incapable of discerning fine shades of gray on high spatial frequencies. The net effect is that some coefficients will require 8 bits to transmit, others few bits, and some can be discarded altogether. Thus effecting a saving in bandwidth.

When the coefficients to be discarded are determined in advance it is called zonal sampling. If the coefficients to be sent are adaptively changed to optimize each picture then it is called adaptive transform coding.

It is relatively simple to extend the concept of a one dimensional transform which operates on a line of a picture, to a two dimensional transform, which transforms the picture in both the horizontal and vertical dimension. The motivation for doing so lies in the fact that elements in a picture are correlated in every direction to the elements around them.

With the picture to be transformed represented as an N x N matrix [X], the two dimensional transform can be carried out by correlating [X] with N^2 orthogonal vectors, each N elements by N elements. These orthogonal vectors $[F]_{k,\ell}$, $k,\ell = 0, 1, 2 \ldots N-1$, comprise the basis pictures of the transform. The result of each correlation is a coefficient, $Y_{k,\ell}$. Mathematically we may write the correlation as:

$$Y_{k\ell} = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} F_{k\ell ij} X_{ij} \qquad (2.3.5)$$

where $F_{k\ell ij}$ is the i,jth term of the k,ℓth vector, and X_{ij} is an element of [X], (see Figure 2).

We may visualize equations 2.3.5 as an operation which places the [X] matrix over one of the $[F]_{k\ell}$ vectors (basis pictures) say $[F]_{2,4}$. The elements that touch each other are then multiplied together and the result of the multiplications are then summed up. This yields coefficient $Y_{2,4}$. The next coefficient is found by using the next $[F]_{k,\ell}$ vector.

We may recover the picture [X], from the coefficients $Y_{k\ell}$ by multiplying each coefficient, $Y_{k\ell}$ by its corresponding basis picture,

$$[F]_{k,\ell} = \begin{bmatrix} f_{k,\ell,0,0} & f_{k,\ell,0,1} & \cdots & f_{k,\ell,0,N-1} \\ f_{k,\ell,1,0} & f_{k,\ell,0.0.} & & \vdots \\ \vdots & \vdots & & \\ f_{k,\ell,N-1,0} & f_{k,\ell,N-1,1} & & f_{k,\ell,N-1,N-1} \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \end{matrix} \Big\} N$$

(a)

$$[X] = \begin{bmatrix} X_{0,0} & \cdots & \cdots \\ X_{1,0} & & \\ \vdots & & \\ \cdot & & \end{bmatrix}$$

(b)

Figure 2 (a) The k,ℓth basis picture shown as an N x N vector

(b) The picture to be transformed represented as an N x N matrix

$[F]_{k\ell}$, and summing up the result. Thus:

$$X_{ij} = \sum_{\ell=0}^{N-1} \sum_{k=0}^{N-1} f_{ijk\ell} \; Y_{k\ell} \qquad (2.3.6)$$

In general, the two dimensional transformation of an N x N picture as given by equation 2.3.5, would require N^4 multiplications and additions, (MA's). Since the N^4 multiplications processes N^2 picture elements, and the sampling rate of the picture elements for standard TV is 125 ns/pel, we must perform N^2 MA's every 125 ns to transform real time TV signals. If the transform technique is to be implemented in hardware, N must be small, and the exponential growth rate of MA's with N must be avoided.

N can be made as small as we wish by dividing up the picture into subpictures, n x n each, and then transforming each n x n subpicture separately. If n is made too small, the ability of the transform to save us bandwidth will suffer. This is because all possible n x n subpictures will become equally likely; therefore, they will correlate equally, on the average, with all the basis pictures, and all the coefficients of equation 2.3.5 will become equally important. Pratt [4] conducted computer simulations with several different types of transforms and his results, shown in Figure 3, suggest that n should be at least equal to 8. The RMS coding error goes down only slightly for $n > 8$ and goes up sharply for $n < 8$. Even with $N = n = 8$, 64 MA's in 125 ns would still be unfeasable.

For certain transforms the n^2 growth rate of MA's can be reduced

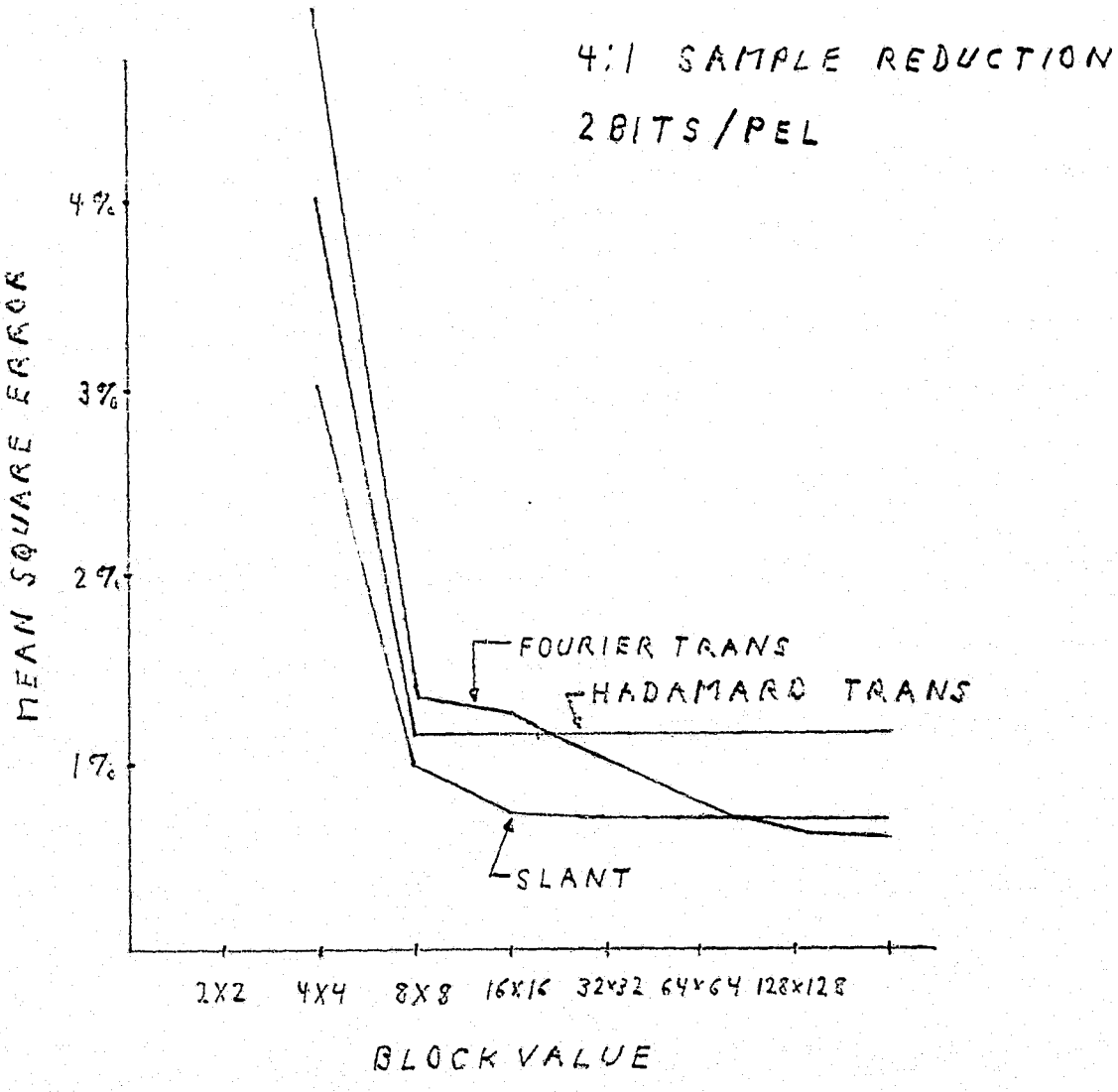


FIG 3 CODING ERROR VS BLOCK SIZE FOR SEVERAL TRANSFORMS

PRATT [4]

to 2n. These transforms are the one's that have the property that their n^2 basis pictures, n x n each, can be derived from n orthogonal vectors of length n each. If as before, we let the k,ℓth basis picture be $[F]_{k\ell}$, then $[F]_{k\ell}$ is formed from the orthogonal vectors $\overline{F}_k$ and $\overline{F}_\ell$ by equation 2.3.7.

$$[F]_{k\ell} = \overline{F}_k \overline{F}_\ell^T \quad K,\ell = 0, 1, 2...n-1 \tag{2.3.7}$$

As an example consider the following:

$$\overline{F}_k = \begin{bmatrix} a \\ b \end{bmatrix} \qquad \overline{F}_\ell = \begin{bmatrix} c \\ d \end{bmatrix}$$

$$[F]_{k\ell} = \begin{bmatrix} a \\ b \end{bmatrix} [c \quad d] = \begin{bmatrix} ac & ad \\ bc & bd \end{bmatrix}$$

If the n orthogonal vectors, used to form the basis pictures, via equation 2.3.7, are arranged to form the rows of a matrix [F], then the two dimensional transform of equation 2.3.5 may be rewritten in matrix notation as:

$$[Y] = \frac{1}{n^2} [F] [X] [F]^T \tag{2.3.8}$$

As before [X]is the picture and [Y] is the matrix containing the transform coefficient.

The inverse transform is given by:

$$[X] = [F]^T [Y] [F] \tag{2.3.9}$$

The series representation for the transform and its inverse is given by equation 2.3.10 and 2.3.11

$$Y_{k\ell} = \frac{1}{n^2} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} X_{ij} F_{ki} F_{j\ell} \tag{2.3.10}$$

$$X_{ij} = \sum_{k=0}^{n-1} \sum_{\ell=0}^{n-1} Y_{k\ell} F_{ik} F_{\ell j} \qquad (2.3.11)$$

It should be obvious to the reader that any one dimensional transform, (as given by equation 2.3.2) with any transform matrix [F], may be expanded to a two dimensional transform via equation 2.3.8. However, not all two dimensional transforms given by equation 2.3.5 can be reduced to equation 2.3.8.

Computation of equation 2.3.10 requires only $2n^3$ MA's. Thus a real time digital encoder using a transform, which could be implemented by equation 2.3.10 would require only 2n MA's every 125 ns. With n = 8 such an encoder would be feasable but still very complicated. Fortunately some of the transforms represented by equation 2.3.8 have transform matrices, [F], that lend themselves to special, fast computational algorithms. The fast Fourier transform (FFT) is probably the best known of these fast computational algorithms. Equation 2.3.10 could be computed with only $2n^2 \log_2 n$ MA's using the FFT algorithm. The real time encoder would then require $2\log_2 n$ multiplications every 125 ns. With n = 8 there would be only 6 MA's every 125ns.

There are four transforms which have fast computational algorithms that have been investigated for picture coding. They are the discrete Fourier-transform, the Harr-transform, the Hadamard-transform, and the slant-transform. From Figure 3 we see that the slant-transform is the best, in the mean square error sense. The slant-transform is the best because several of its basis vectors look very much like a typical line

of a picture. If several basis vectors correlate strongly with the picture then the rest must correlate weakly since the vectors are orthogonal. The weak ones can be discarded with little effect on the reconstructed picture.

The basis vectors of the slant-transform are shown in Figure 4. The first vector is the constant valued vector. This vector correlates most strongly with pictures because a large number of lines in a picture are of constant grey level over a considerable length. Another typical image line is one which increases or decreases in brightness in a linear fashion [4]. Such a line would correlate well with the second basis vector. The rest of the basis vectors correlate poorly with typical image lines. Figure 5 shows the number of bits allotted by Pratt [4] to each element in the coefficient matrix. (Matrix [Y] in equation 2.3.8.) He reports excellent subjective quality pictures using his slant-transform at 1.5 bits/pel for his computer simulations.

After all is said and done about transform coding, it turns out that such encoders are very complex and not practical for most real time applications. As of 1977 only one such encoder has ever been built, and it used the Hadamard-transform. The Hadamard-transform is particularly attractive for real time operation because its basis vectors contain only plus and minus 1's. With ±1's as the elements of [F], and a fast computational algorithm as well, equation 2.3.8 can be computed with $2n^2\log_2 n$ additions and no multiplications.

The Hadamard-transform matrix is shown on the next page for n = 4

8	8	8	7	7	7	5	5	4	4	4	4	4	4	4	4
8	8	7	5	5	5	3	3	3	3	3	3	3	3	3	3
8	7	6	4	4	4	3	3	2	2	2	2	2	2	2	2
7	5	4	3	2	2	2	2	0	0	0	0	0	0	0	0
7	5	4	2	2	2	2	2	0	0	0	0	0	0	0	0
7	5	4	2	2	2	2	2	0	0	0	0	0	0	0	0
5	3	3	2	2	2	0	0	0	0	0	0	0	0	0	0
5	3	3	2	2	2	0	0	0	0	0	0	0	0	0	0
4	3	2	0	0	0	0	0	0	0	0	0	0	0	0	0
4	3	2	0	0	0	0	0	0	0	0	0	0	0	0	0
4	3	2	0	0	0	0	0	0	0	0	0	0	0	0	0
4	3	2	0	0	0	0	0	0	0	0	0	0	0	0	0
4	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0
4	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0
4	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0
4	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0

FIG 5 NUMBER OF BITS ASIGNED TO EACH COEFFICENT OF THE SLANT TRANSFORM

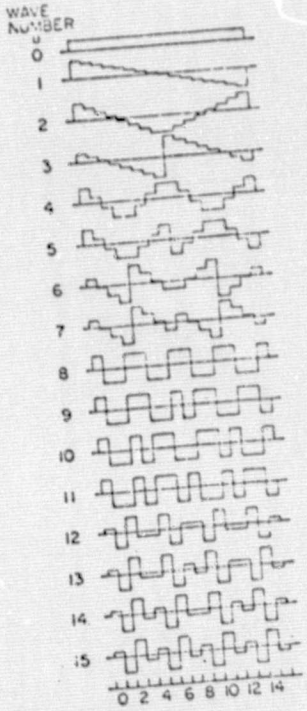


FIG 4 SLANT TRANSFORM VECTORS

PRATT (4)

$$H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \qquad (2.3.12)$$

In 1976 Linkabit Corporation implemented a real time Hadamard transform scheme which reduced the number of bits per picture element (pel) from 6 bits for PCM to 2 bits for the Hadamard-transform scheme (5). The scheme used by Linkabit is approximately the same as the one desscribed below.

The Linkabit scheme subdivided a 512 by 512 sampled picture into 16,384 subpictures of 4 x 4 samples each. Each one of the subpictures [X] was transformed into a 4 x 4 matrix [Y] via Equation 2.3.8 and 2.3.12. Of the 16 elements of [Y] shown below

$$[Y] = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & Y_{14} \\ Y_{21} & Y_{22} & Y_{23} & Y_{24} \\ Y_{31} & Y_{32} & Y_{33} & Y_{34} \\ Y_{41} & Y_{42} & Y_{43} & Y_{44} \end{bmatrix} \qquad (2.3.12)$$

Y_{22}, Y_{23}, Y_{24}, Y_{32}, Y_{42} and Y_{44} were discarded, i.e. set to zero. The remaining elements of [Y] were quantized as follows:

Y_{11} is quantized by 64 levels (i.e. 6 bits)

Y_{12} and Y_{21} are quantized by 8 levels (i.e. 3 bits)

Y_{13} and Y_{31} are quantized by 16 levels (i.e. 4 bits)

Y_{14} and Y_{41} are quantized by 9 levels (i.e. 3.2 bits)

Y_{33} and Y_{34} and Y_{43} are quantized by 5 levels (i.e. 2.3 bits)

Thus 33 bits were required to encode the [Y] matrix. This is about 2 bits per picture element. (By taking advantage of interframe redundancy, the actual scheme used by Linkabit further reduced the number of bits per pel to 1).

It is reported by Linkabit that the encoder produced good pictures at the 2 bit per pel rate.

2.4 Predictive Encoding

Another class of encoders, called predictive encoders, is being studied for use in compressing the bandwidth of digitized video signals.

A linear predictive encoder is defined by the following two equations

$$X_o = a_1 S_1 + a_2 S_2 + \ldots a_n S_n \qquad (2.4.1)$$

$$e_o = S_o - X_o \qquad (2.4.2)$$

S_o, S_1, S_2, ... S_n are successive samples of the video signal. S_o is

the current sample. X_o is the predictors estimate of S_o, based upon previous samples S_1, S_2, ... S_n, and e_o (the error in the prediction) is the signal sent to the receiver. a_1, a_2, ... a_n are constants chosen to minimize e_o. If e_o is small, i.e. X_o is a good estimate of S_o, then bandwidth compression results because, fewer bits would be required to transmit e_o than would be required to transmit S_o. The receiver can reconstruct the picture element, S_o, by computing $X_o + e_o = S_o$.

The constants a_1, a_2, ... a_n can be found from linear prediction theory in a straightforward manner by assuming that minimizing the mean square error between the original picture and the encoded picture is a satisfactory measure of the quality of the picture. The mean square error is given by

$$E[(S_o - X_o)^2] = E[(S_o - (a_1S_1 + a_2S_2 + \ldots a_nS_n))^2] \quad (2.4.3)$$

where $E[\ ]$ denotes expected value.

To find the minimum we take the partial derivatives of $E[(S_o - X_o)^2]$ with respect to each one of the a's and set them equal to zero

$$\frac{d}{da_i} E[(S_o - X_o)^2] = -2E[(S_o - (a_1S_1 + a_2S_2 + \ldots a_nS_n))S_i] = 0$$

$$E(S_o,S_i) = a_1E(S_1,S_i) + a_2E(S_2,S_i) + \ldots a_nE(S_n,S_i) \quad (2.4.4)$$

If we represent the covariance of S_i and S_j by

$$R_{ij} = E[S_i S_j]$$

then we can rewrite equation 2.4.4 as

$$R_{oi} = a_1 R_{1i} + a_2 R_{2i} + \ldots + a_n R_{ni} \qquad (2.4.5)$$

Equation 2.4.5 defines a set of n simultaneous linear equations in the n unknowns a_i, i = 1,2, ..., n, which can be found if the covariance R_{ij} are known.

Equations 2.4.1 and 2.4.2 are implemented by a recursive relationship as shown in equations 2.4.6 and 2.4.7

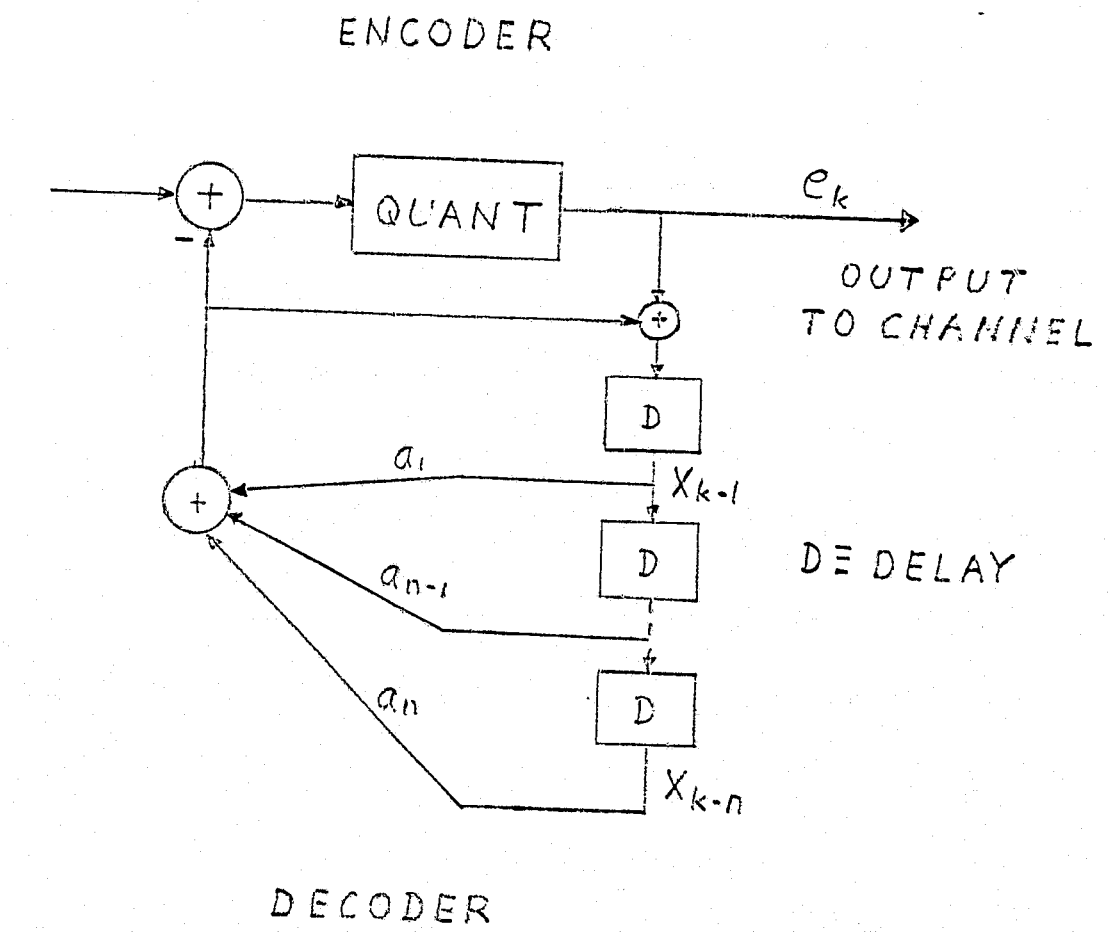
$$X_k = a_1 X_{k-1} + a_2 X_{k-2} + \ldots a_n X_{k-n} + e_k \qquad (2.4.6)$$

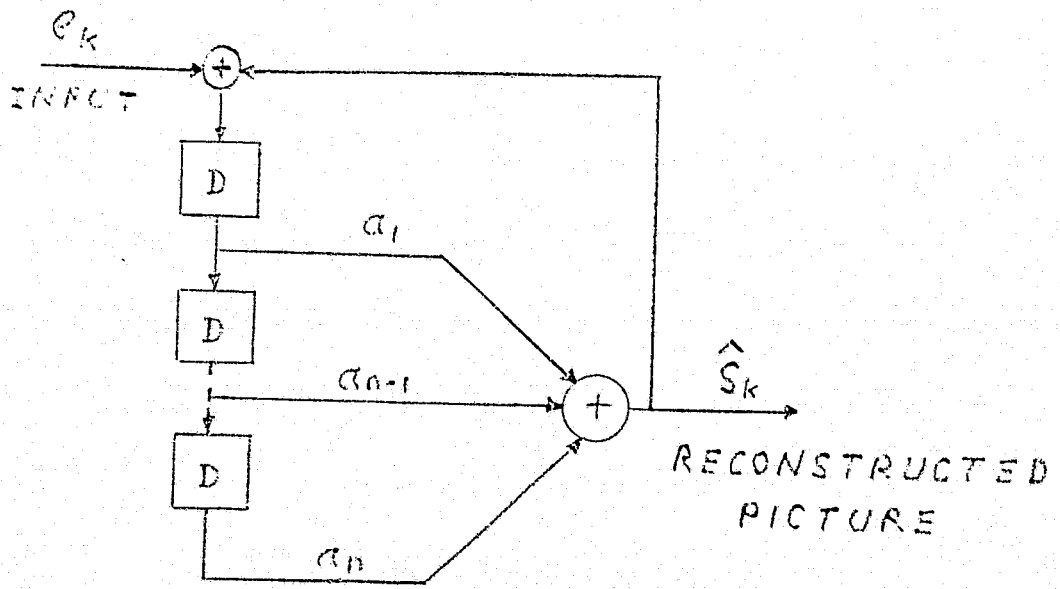
$$e_{k+1} = \text{QUANT}[S_{k+1} - X_k] \qquad (2.4.7)$$

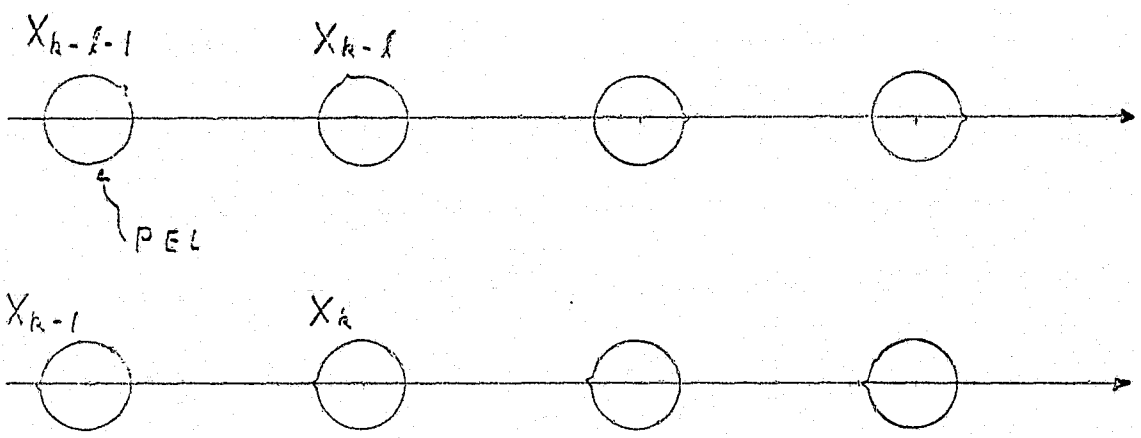
The function, QUANT[ ], defines a non-linear quantizer. An implementation diagram for equation 2.4.6 and 2.4.7 is shown in Figure 6.

A number of investigations have been carried out to determine an optimal set of a_i's for equation 2.4.5, and an optimal quantizer for equation 2.4.7 (6,7,8). It was first determined that any given pel of a picture is correlated only to the pels adjacent to it. With reference to Figure 7, which labels the pels in the picture array, we can see that

Fig. 6 Predictive Coding







ℓ = NUMBER OF PELS PER LINE

FIG 7 THE TWO DIMENSIONAL ARRAY OF PICTURE ELEMENTS (PELS)

there are only four pels adjacent to X_k which can be used in equation 2.4.6. In practice it has been found that the three pels, $X_{k-\ell-1}$, $X_{k-\ell}$, and X_{k-1} are sufficient. The set of a_i's that has been found to give the best subjective results is shown in equation 2.4.8.

$$X_k = .75X_{k-1} - .5X_{k-\ell-1} + .75X_{k-1} \qquad (2.4.8)$$

From O'Neal's work (6) it can be determined that the optimum 8 level quantizer (3 bits/pel) for a picture requiring 128 brightness levels would have a transfer function as shown in Figure 8.

Figures 7 and 8 and equation 2.4.8 describe the best non-adaptive, two dimensional, 3 bit per pel differential PCM, (DPCM), encoder developed. (It is non-adaptive because the quantizer levels are fixed and it is two dimensional because it uses correlation in both the vertical and horizontal directions.)

A comparison between the DPCM predictive encoder and the two-dimensional transform encoder reveals that the DPCM encoder is less complex than the transform encoder, but the DPCM encoder requires more bits per pel. I would estimate that the transform encoder would require about one order of magnitude more IC's to implement as compared with the DPCM encoder. Wintz' reports that transform encoding at 2 bits per pel results in pictures whose quality is similar to DPCM encoded pictures at 3 bits per pel.

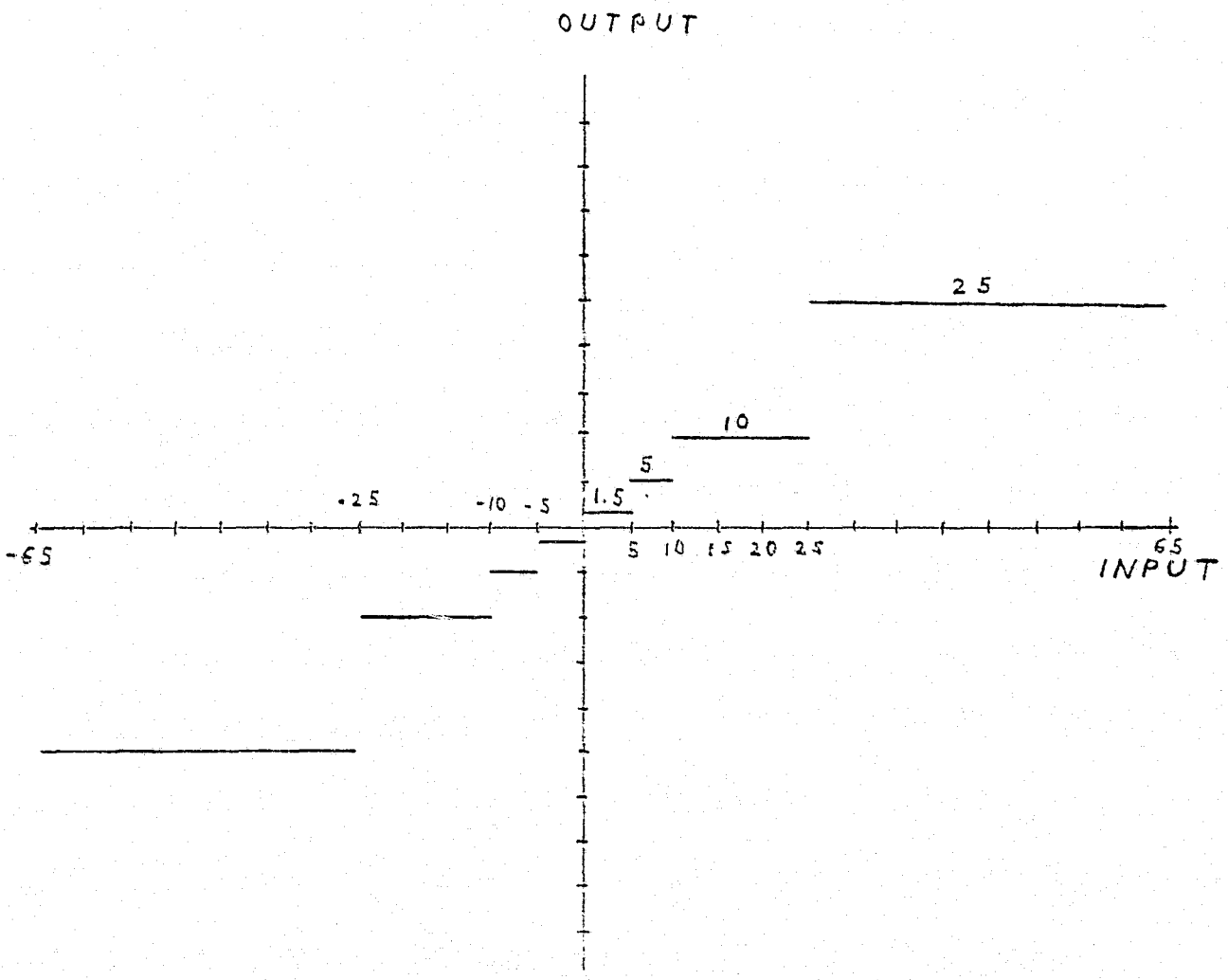


FIG 8 TRANSFER FUNTION OF 3 BIT QUANTIZER OF PREDICTIVE ENCODER

2.5 Interframe Encoding

Except for sudden scene changes successive frames of a TV picture are nearly identical. Studies done on broadcast TV signals reveal that on the average only 10% of the picture elements change by more than 2% between successive frames of a TV signal[9]. This high correlation between successive frames, can be used to reduce the bandwidth of TV signals using techniques similar to the techniques discussed previously.

Transform coding and predictive encoding are adapted to interframe encoding by treating picture elements displaced along the time axis (same pel but in successive frames) the same way as pels displaced along the horizontal or vertical axis.

Another approach, unique to interframe encoders, uses the difference signal between frames as the basic element in the interframe coder. In this scheme the previous frame stored in a frame memory is compared sample by sample to the sample values of the present frame. If the difference between samples from successive frames exceeds a given threshold that sample value is replenished by transmitting it and updating the frame memory. It is also necessary to transmit the address of the replenished sample. The number of replenished samples transmitted to the receiver is a function of the amount of motion in the scene. A large buffer is required to average out the transmitted bit rate. For picturephone application this technique can reduce the bit rate to 1 bit per pel[10]. Broadcast TV would require at least 2 bits per pel.

2.6 Conclusions

Transform coding, predictive encoding, or interframe encoding can reduce the number of bits per pixel from 6 for PCM to 2 or 3 with only slight degradation of the original picture.

It is not possible to say that one type of encoder is superior to another for each can be made to seem the best (or worst) given the proper tradeoffs. The simplest encoders to implement are the predictive encoders while the most complex are the interframe encoders. Although the interframe encoders are very complex, in the near future they may become inexpensive because of the decreasing cost of the frame storage used in interframe encoding.

CHAPTER III

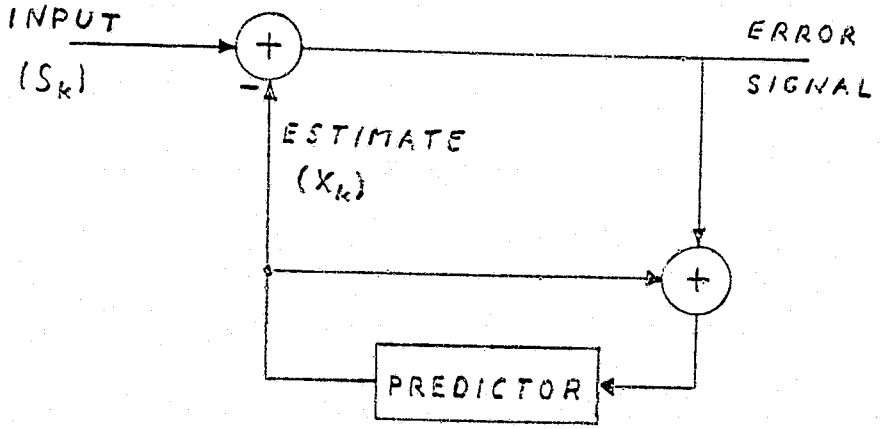
A Delta Modulator for Encoding Video Signals

After studying the various encoding schemes mentioned in Chapter II we decided to implement our video encoder using predictive encoding. We further decided to restrict ourselves to the class of predictive encoders known as adaptive delta modulators (ADM).

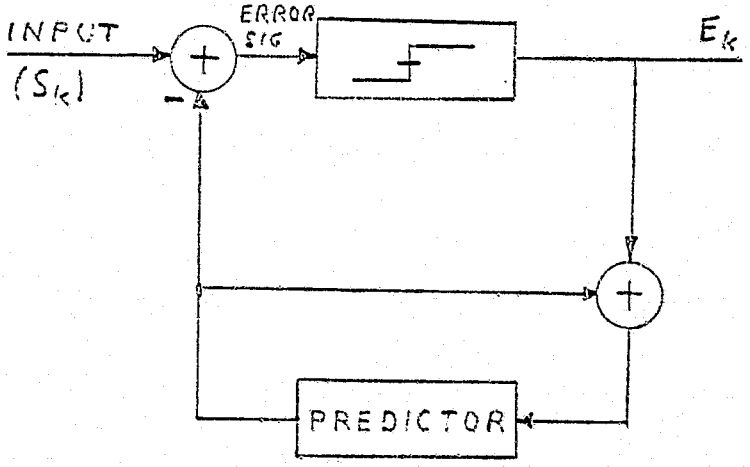
Figure 9 shows in block diagram form the evolution of the ADM from the linear predictive encoder. The linear predictive encoder of Figure 9a will generate an estimate signal, (X_k), that is a perfect replica of any input signal, (S_k), provided that the number of bits used to represent the error signal is unrestricted. In the ADM of Figure 9b the error signal passes through a hard limiter and is thus restricted to the values ± 1. If the predictor box of Figure 9b could always predict the value of the input (S_k) to within ± 1 quantization level then the ADM would also produce an estimate that is a perfect replica of the input signal. Furthermore, the ADM would have achieved a bandwidth compression of 6 to 1 over 6 bit PCM since only one bit, $(E_k = \pm 1)$, is sent to the receiver for each sample of the input signal.

Our experience with delta modulators has shown that for sampling rates that result in bandwidth savings the predictor will not predict the input signal to within ± 1 quantization level; therefore, another predictor, the step size generator, has been added to Figure 9c. The step size generator attempts to reconstruct the error signal from the E_k's. The reconstructed error signal is called the step size (Y_k).

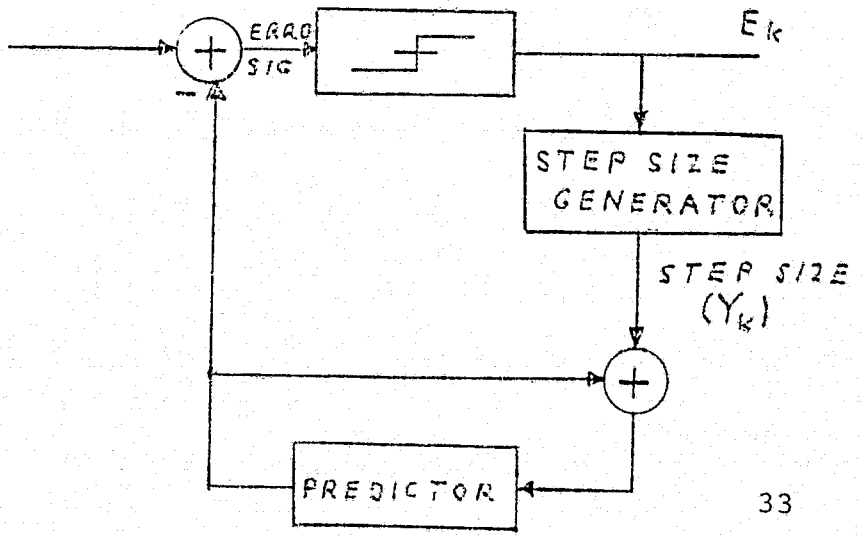
FIG 9 EVOLUTION OF THE DELTA MOD.



(a)



(b)



(c)

As of now, no general mathematical technique exists to find the optimum step size generator given the psychophysics of vision and the statistics of video signals. Step size generators are chosen by a combination of intuition, mathematics, and trial and error. In 1971 Song, Schilling, and Garodnic proposed a video delta modulator based upon a video signal model that was first order Markov[11]. The rest of this chapter gives the results of computer simulations of the ADM and the results of a hardware implementation of the ADM.

3.1 The Delta Modulator

The delta modulator chosen to encode the pictures in this paper is shown in Figure 10. The equations describing the operation of the delta modulator are given below:

$$E_{k+1} = \mathrm{Sgn}(S_{k+1} - X_{k+1}) \tag{3.1.1}$$

$$X_{k+1} = X_k + Y_{k+1} \tag{3.1.2}$$

$$Y_{k+1} = \begin{cases} |Y_k|(aE_k + bE_{k-1}); & |Y_k| \le Y_{max}/1.5,\ E_k = E_{k-1} \\ & |Y_k| < Y_{max},\ E_k \neq E_{k-1} \\ & 2Y_{min} \le |Y_k| \end{cases} \tag{3.1.3a}$$

$$Y_{k+1} = 2Y_{min}E_k \quad ; \quad |Y_k| < 2Y_{min} \tag{3.1.3b}$$

$$Y_{k+1} = Y_{max} \quad ; \quad |Y_k| > Y_{max}/1.5,\ E_k = E_{k-1} \tag{3.1.3c}$$

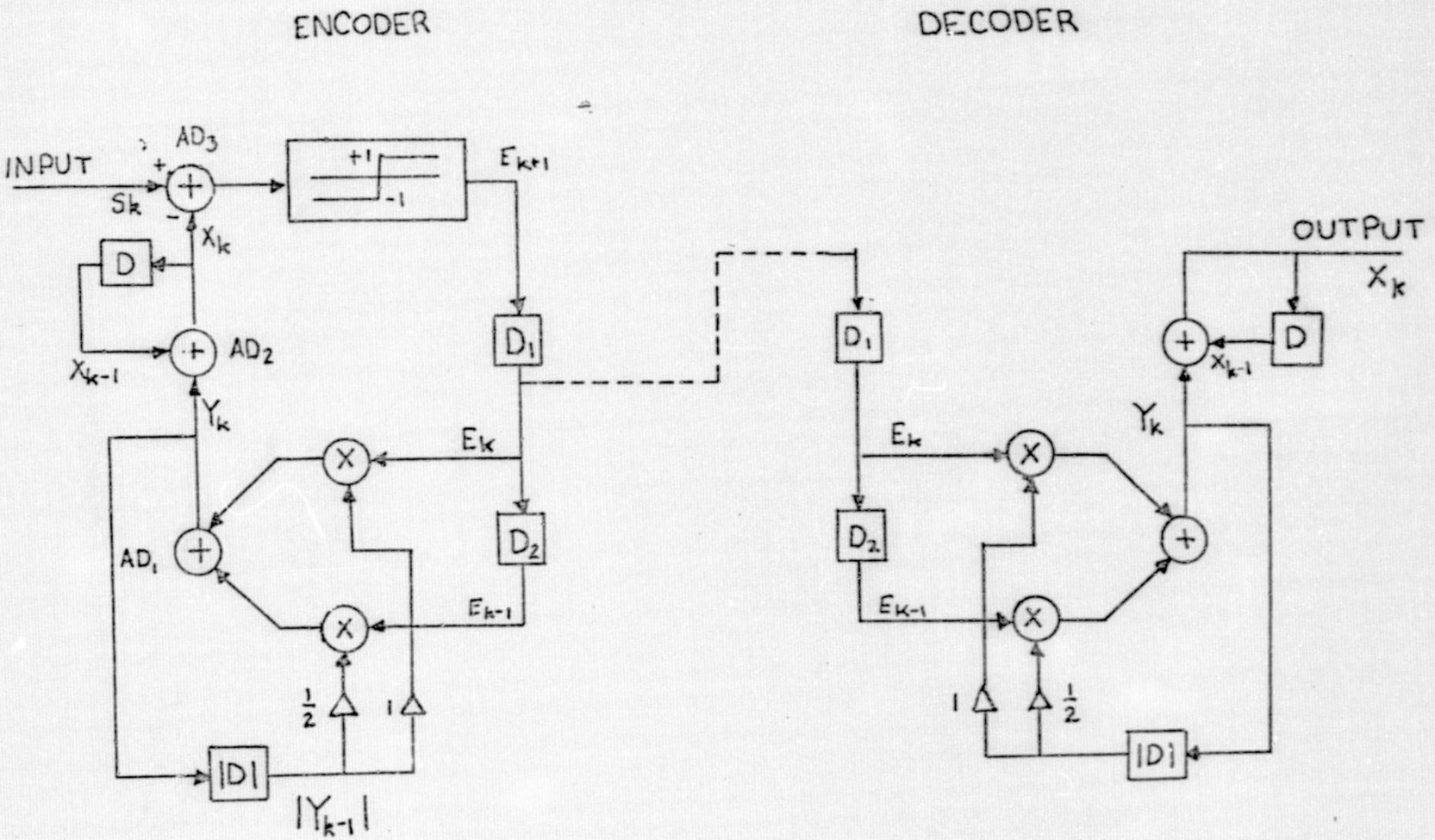


FIG. 10 DIGITAL ADAPTIVE DELTA MODULATOR

E_k = Output of the encoder
S_k = Input to the encoder
X_k^k = The encoder's estimate of the input signal at the kth instant of time; also, the decoder's output at the kth instant of time
Y_k = The step size of the delta modulator
Y_{min}^k, Y_{max} = Constants that determine the minimum and maximum allowable value for Y_k
a,b = Other constants

There are other delta modulator algorithms besides the one described by equations 3.1.1, 3.1.2, 3.1.3. In general the differences lie in equation 3.1.3, the way the step size is formed. In some delta modulators $|Y_k|$ is a constant. In others it increases or decreases linearly. In our delta modulator the step size, Y_k, changes exponentially. It has been experimentally determined that an exponentially changing step size type of delta modulator is more appropriate than the other types of delta modulators at encoding a signal with large step type changes in amplitude as found in video signals.

In order to test our delta modulator on video signals from actual pictures, and to find good values for a, b, Y_{min} and Y_{max} we set up a computer processor for pictures as shown in Figure 11. With the experimental apparatus of Figure 11 we were able to program the PDP 8 computer to simulate a delta modulator. By varying the values of a, b, Y_{min} and Y_{max} in the computer program, and then comparing the resulting pictures, we obtained values for a, b, Y_{min} and Y_{max} which would produce satisfactory pictures.

In choosing the best set of values for a, b we also considered the effects of different sets of values on a real time hardware

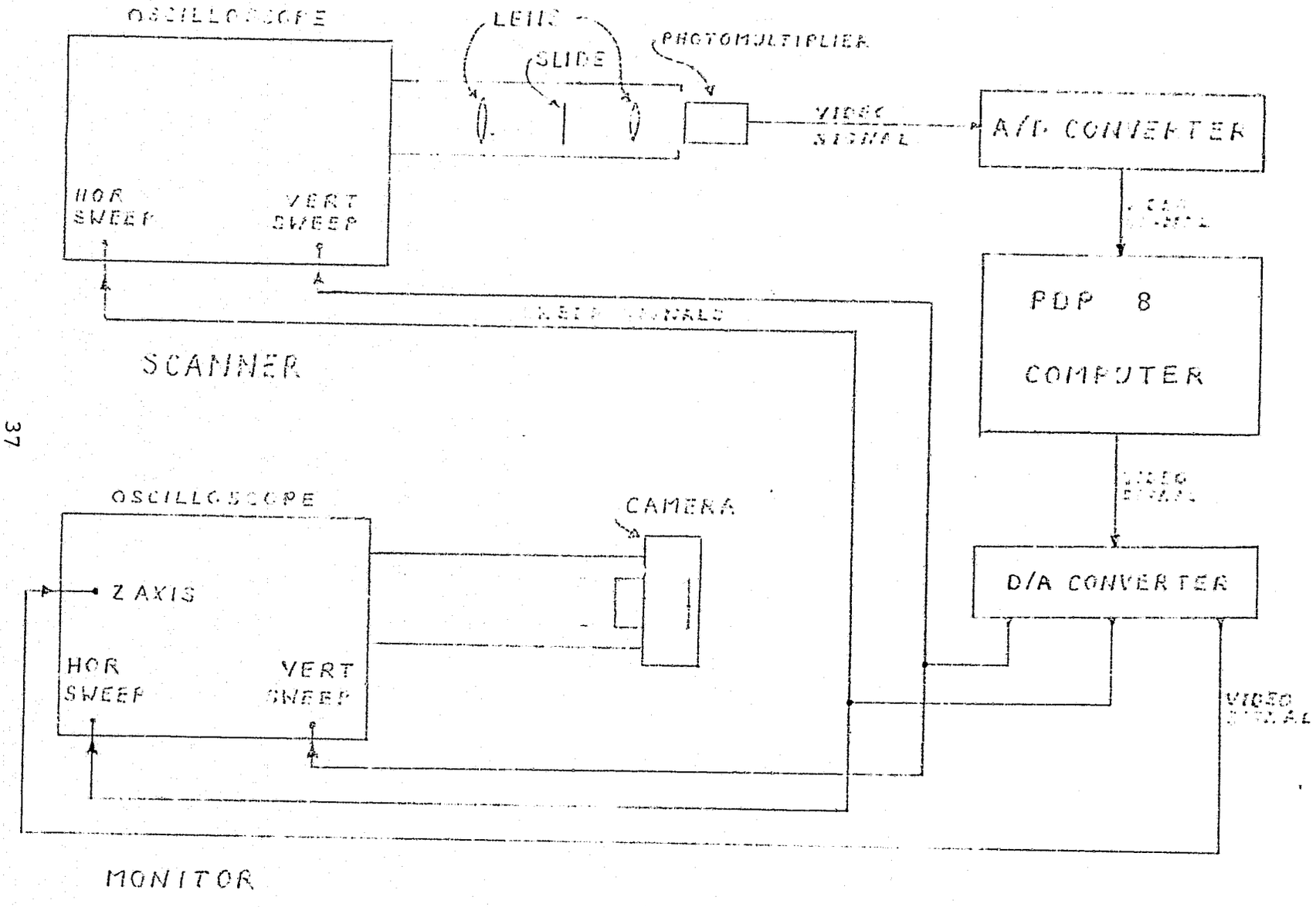


FIG. 11 COMPUTER PROCESSOR FOR PICTURES

implementation of the delta modulator. Since very high sampling rates would be required for the real time processing of video signals, we restricted our choice of values of a and b to those values that would minimize the time required to perform the multiplications of $|Y_k|$ by a and b. From the computer simulations we found that a = 1 and b = .5 produce about the best pictures. Since these values are also powers of 2 the multiplication by 1 and .5 are wired shift operations and require no hardware to implement nor any time to perform. Clearly, a = 1 and b = .5 was an ideal choice for a and b and thus these values were used in all our delta modulators.

To determine the effects of Y_{min} and Y_{max} on a delta modulated encoded picture we examined the response of the delta modulator to a step like input. We chose a step like input because the edges of objects in pictures produce step like changes in the video signal. From Figure 12 we see that at first the delta modulator's output does not rise as fast as the input signal. This effect gives rise to a type of degradation called slope overload noise. After the delta modulator catches up with the input signal it may overshoot the input, and then after a period of time known as the settling time, the delta modulator settles down to a repetitive four bit pattern that bounces around the input signal. This last effect is a type of degradation called the granular noise. The amplitude of the granular noise determines the minimum voltage change of the input signal that can be resolved by the delta modulator. Note also that when the step size, Y_k, increases,

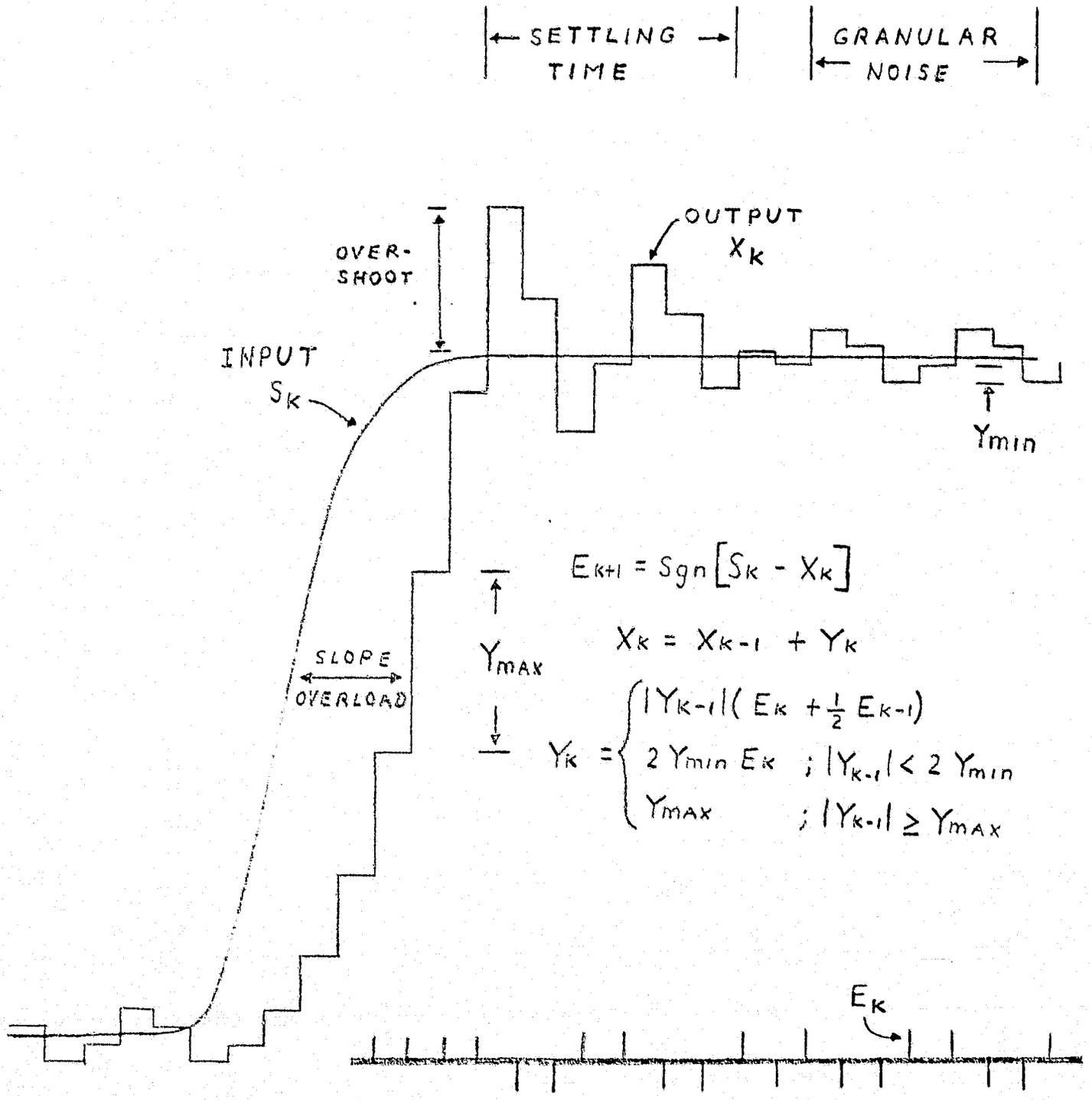


Fig. 12 The response of the delta modulator to a step like input

it increases by a factor of 1.5 until Y_{max} is reached. Anytime the step size decreases, it will decrease by a factor of .5 or until Y_{min} is reached. This follows from equation 3.13 and a = 1, b = .5.

Table 1 summarizes the relationship between slope overload noise, granular noise, overshoot, settling time and increasing or decreasing the value of Y_{min} and Y_{max}. From Table 1 it is clear that a tradeoff exists between slope overload noise and granular noise when choosing a value for Y_{min}. For Y_{max} the tradeoff is between slope overload noise against settling time and overshoot amplitude. From our computer simulations on real pictures we found that a value for Y_{min} equal to 1/64 the peak to peak video input signal produced the best pictures. This result is not surprising since this value of Y_{min} will allow the delta modulator to resolve 64 gray levels and it is well known that at least 64 gray levels are required to produce satisfactory pictures. It was also found that the quality of our pictures was not very sensitive to the value of Y_{max}. It was observed that the value of Y_k rarely grew larger than 1/4 the peak to peak input signal and that large overshoots and ringing usually produced out of band frequency components that were eliminated by low pass filtering. After considering all factors we chose a value of 1/4 the peak to peak input signal for Y_{max}.

In conclusion we found that the delta modulator worked best with a = 1, b = .5, Y_{min} = 1/64 p-p input signal and Y_{max} = 1/4 p-p input signal. These values were used in all subsequent computer simulations.

	SLOPE OVERLOAD NOISE	GRANULAR NOISE	SETTLING TIME	OVERSHOOT AMPLITUDE
Increasing Y_{min}	Decreases	Increases	No Change	No Change
Decreasing Y_{min}	Increases	Decreases	No Change	No Change
Increasing Y_{max}	Decreases	No Change	Increases	Increases
Decreasing Y_{max}	Increases	No Change	Decreases	Decreases

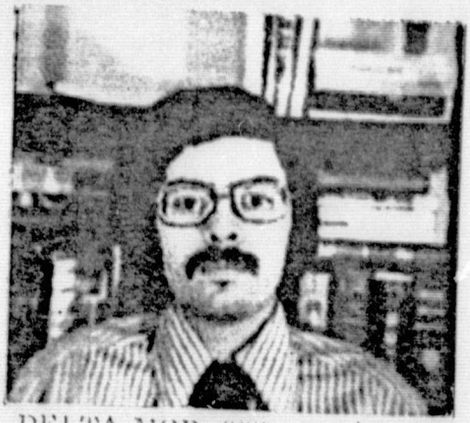
TABLE I

3.2 Computer Simulation of Delta Modulated Pictures

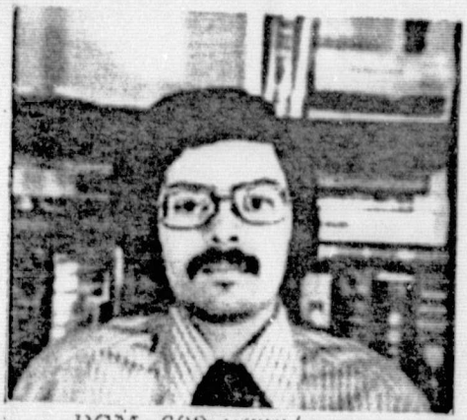
Using the experimental apparatus shown in Figure 11 we generated the pictures shown in Figure 13. The delta modulated pictures were taken with the optimum parameters given in section 3.1 and the PCM encoded pictures were encoded with 6 bits per sample.

If we examine the PCM encoded pictures, we notice that as the number of bits per line decreases so does the resolution of the picture. The top picture has 684 bits per line which means that at 6 bits per sample this picture contains 114 samples per line. Thus the smallest resolvable object or black to white transition must extend for 1/114 of a line length. The bottom PCM picture has only 68 samples per line; hence, the smallest resolvable object is 5/3 times as large, and black to white transitions are 5/3 times longer. Using standard terminology we would say that one picture has 114 picture elements (pels/line) while the other has 68 pels/line. For comparison, a standard TV picture on a good studio monitor has about 500 pels/line.

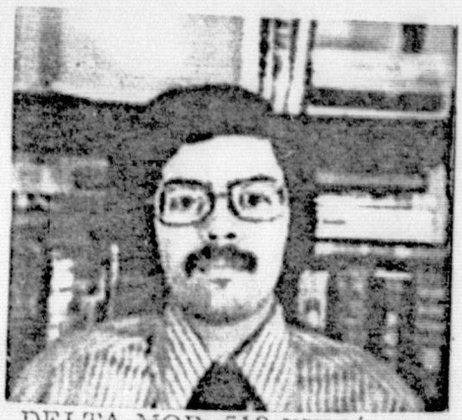
The one to one relationship between the number of samples per line and pels (which is measure of resolution) holds only if the video signal is sampled at the Nyquist rate. If the picture is sampled at the Nyquist rate (twice the highest frequency component) then each sample is a pel. If the video signal is sampled above the Nyquist rate no increase in resolution will result, rather one would have several samples of the same pel. Sampling below the Nyquist rate results in a



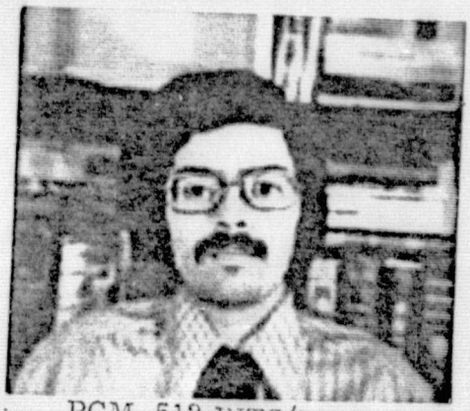
DELTA MOD 682 BITS/LINE



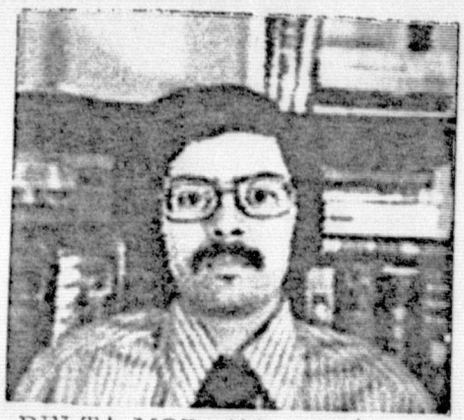
PCM 682 BITS/LINE



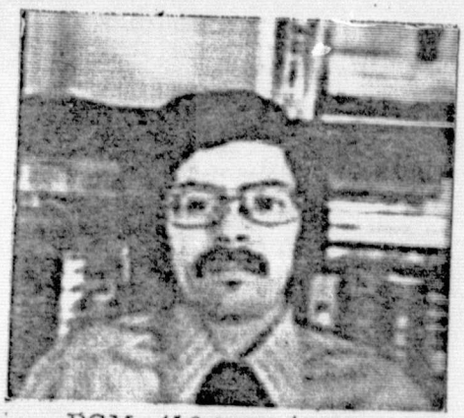
DELTA MOD 512 BITS/LINE



PCM 512 BITS/LINE



DELTA MOD 410 BITS/LINE



PCM 410 BITS/LINE

Fig. 13 A comparison of an adaptive delta modulator and a PCM encoder of video signals; 170 scaning lines per picture; PCM pictures have 64 quantization levels.

loss of resolution and the production of annoying Moire patterns. (Video equivalent to aliasing).

The PCM pictures of Figure 13 were always sampled at the Nyquist rate. This was accomplished by low pass filtering the video signal before it was sampled so that the sampling rate would always be the pel (Nyquist) rate. The resulting samples were then low pass filtered to construct a continuous picture from the sampled pictures.

The pictures on the left side of Figure 13 show the results of delta modulating a picture at various bit rates. All three pictures were filtered so that they had 114 pels per line before encoding. Since a delta modulator transmits one bit per sample, the number of bits per line is also the number of samples per line. The number of bits per pel is given by the number of bits per line divided by the number of pels per line.

As the number of bits per line decreases, the delta modulated pictures degrade in a manner quite different from the PCM picture. At 684 bits/line, 6 bits/pel, the delta modulated picture looks the same as the PCM picture. At 408 bits/line the PCM picture has blurred while in the delta modulated picture edges begin to wiggle. We call this edge busyness.

Edge busyness can be understood by referring to Figures 14 and 15. Figure 14a shows a vertical edge before it is delta modulated and Figure 14b shows the same edge after it is delta modulated. The wiggliness of the edge can be understood by referring to Figure 15. The solid

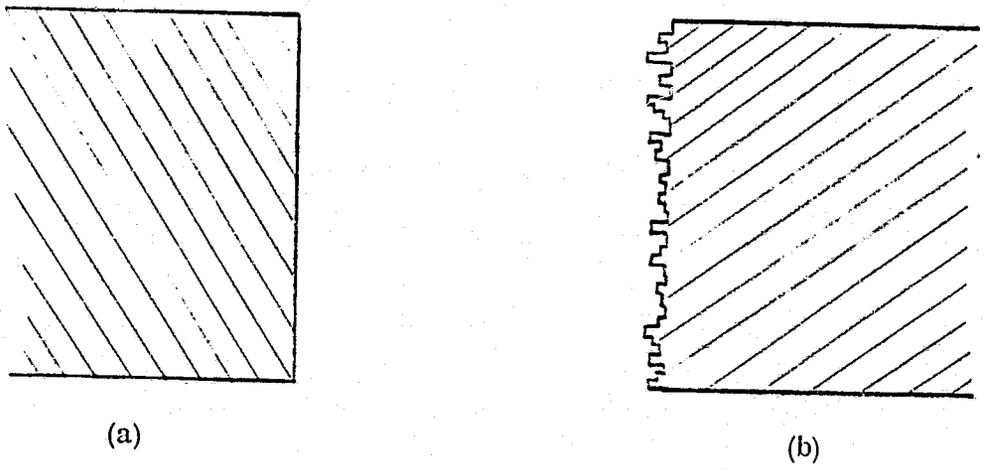


Fig. 14. The effects of edge business on an edge
(a) Edge before delta mod encoding
(b) Edge after delta mod encoding

line in Figure 15 shows the response of the delta modulator to step input S_k. The dotted line shows the response of the same delta modulator to the same step input but the delta modulator had different initial conditions at the start of the step. In the case of the solid line the delta modulator acquired the amplitude of the step in 7 sample times while in the case of the dotted line it took only 5 sample times. In a picture this would have the effect of delaying an edge at some locations, but not at others. The resulting effect is to cause all sharp edges at right angles to the direction of scan to wiggle. This effect is edge busyness.

3.3 Results of the Computer Simulations

The computer simulations provided two important results. First it gave us the values of Y_{min}, Y_{max}, a and b for an adaptive delta modulator which could be used to encode video signals. The second result was that it revealed that at low bit rates the delta modulator will degrade pictures by producing edge busyness. At 6 bits per pel the edge busyness is not noticable. At 3 bits per pel, edge busyness is noticable but there is almost no loss in resolution. This last result was encouraging since the delta modulator would have to operate at around 3 bits per pel to achieve a bandwidth compression factor equivalent to the other bandwidth compression techniques described in Chapter 2.

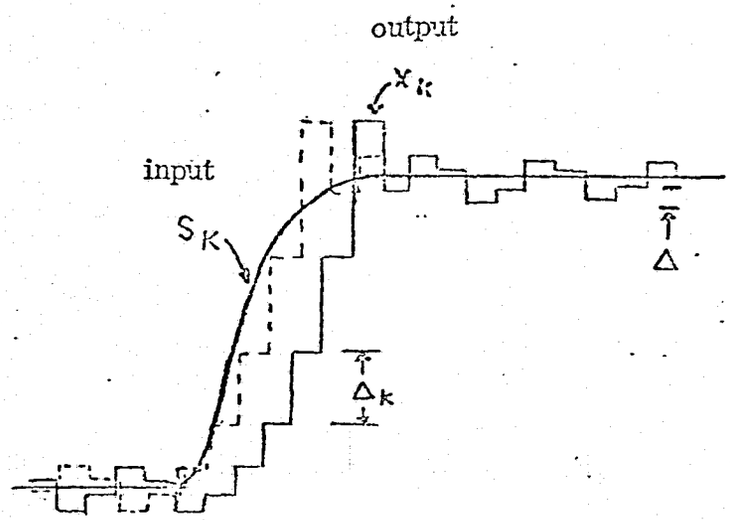


Fig. 15 · Response of the Delta Modulator to a Step Input with Different initial Conditions

3.4 A Real Time Delta Modulator

Although computer simulations are useful they can often be misleading. A still picture 2 inches by 2 inches with a resolution of only 114 pels may not show up all types of defects that would be visible on a standard TV monitor displaying a motion picture. To investigate the behavior of the delta modulator on motion pictures displayed on a standard TV monitor it became necessary to build a high speed delta modulator which could encode pictures from an NTSC standard TV camera.

The block diagram for the high speed video delta modulator is shown in Figure 16. This configuration was picked because it would maximize the sampling rate of the delta modulator when implemented with standard Schotky TTL IC's. The operation of Figure 16 will be explained by tracing the processing of the video signal through each block with reference to equations 3.1.1 to 3.1.3. To implement the equation 3.1.1 the analog video signal from the TV camera is applied to one input of a comparitor. The estimate (X_{k+1}) is at the other input. The comparator takes the sign of the difference between the two signals and outputs E_{k+1}. The two flip-flops on the top of the diagram store E_k and E_{k-1}. The upper adder/subtractor implements the first part of equation 3.1.3. The inputs to the adder/subtractor are the step size, $|Y_k|$ and $|Y_k|$ with each bit displaced one bit to the right. Thus, when the adder/subtractor adds, the result is:

$$|Y_k| = |Y_k| + \tfrac{1}{2}|Y_k| = 1.5|Y_k|$$

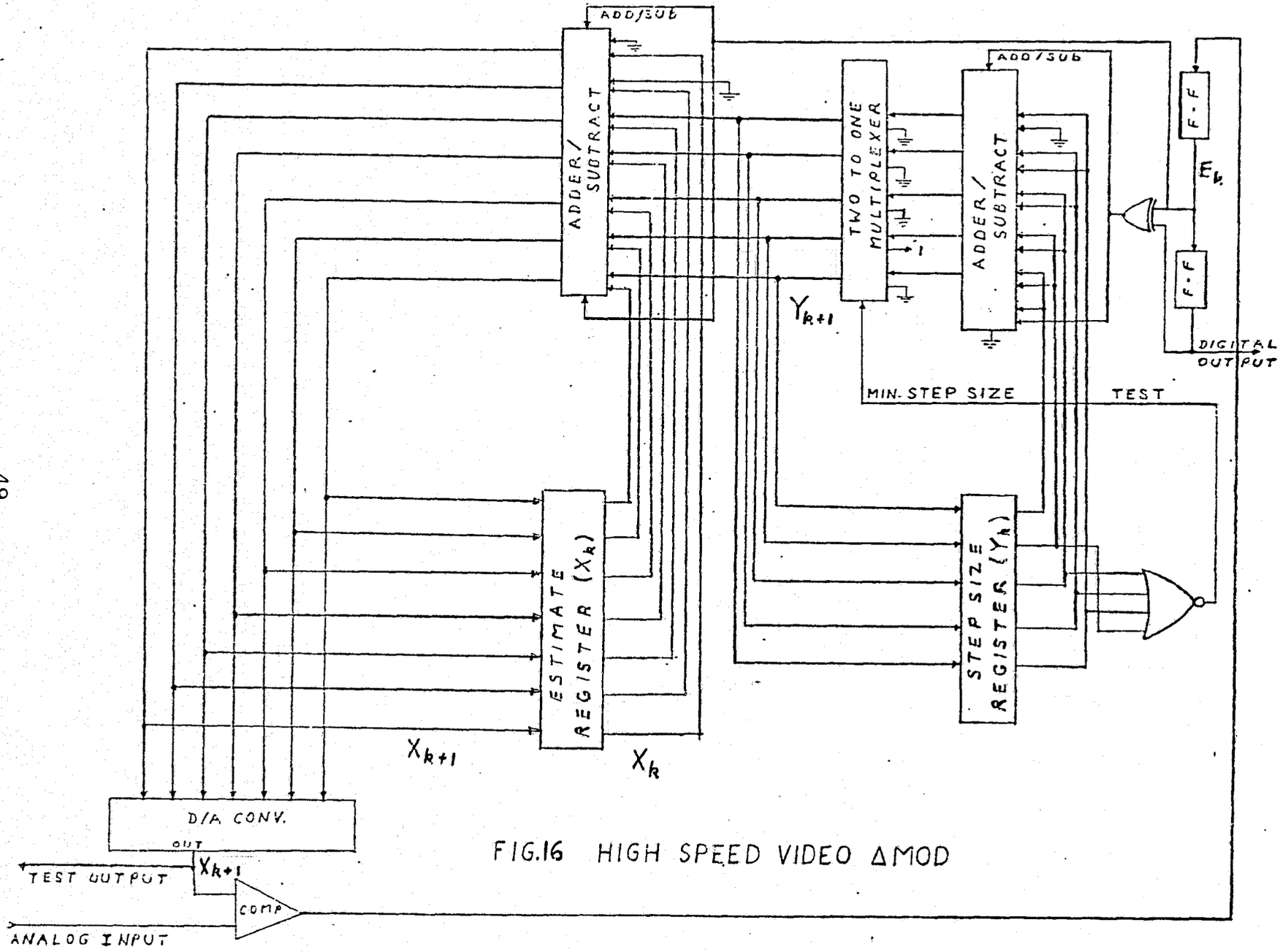


FIG.16 HIGH SPEED VIDEO ΔMOD

and when the adder/subtractor subtracts, the result is:

$$|Y_{k+1}| = |Y_k - \tfrac{1}{2}Y_k| = .5|Y_k|$$

The output of the exclusive or gate signals the adder/subtractor to add when E_k and E_{k-1} are of the same sign and to subtract when E_k and E_{k-1} are of different sign. This is equivalent to saying that the absolute value of the new step size is increased by a factor 1.5 whenever two E_k's in a row are of the same sign and decreased by a factor of .5 whenever two consecutive E_k's are of opposite sign.

The "nor" gate and multiplexor are used to implement the second part of equation 3.1.3, the minimum step size test. The "nor" gate's output goes to logic one if $|Y_k|$ becomes too small, i.e., falls below the number 2. This signals the multiplexor to output $|Y_{k+1}|$ as the number 2. Otherwise, the multiplexor outputs $|Y_{k+1}|$ from the adder/subtractor.

The lower adder/subtractor implements equation 3.1.2. It adds or subtracts the absolute value of the step size, $|Y_{k+1}|$, to the estimate, X_k. It adds when E_k is +1 and subtracts when E_k is -1. It actually performs the operation

$$X_{k+1} = X_k + E_k|Y_{k+1}|$$

but this is equivalent to equation 3.1.2 because the sign of Y_{k+1} is always the same as E_k.

Several parameters that we used in the computer simulations were changed in the hardware implementation of the delta modulator. The number of quantization levels used to represent the estimate in Figure 16 is 128 (7 bits) whereas we used 64 quantization levels in the computer simulations. It was felt that the extra quantization levels would be necessary to accomodate the negative going sync pulses which are part of the composite video signals coming from the TV camera. The extra levels would also prevent the estimate register from overflowing on very bright areas of the picture. We used 31 as the maximum step size (5 bit arithmetic) with a provision to reduce it to 15 (4 bit arithmetic) to see which would give better results. If the results were the same, future delta modulators would be built with four bit arithmetic in the step size section. The additional adder/subtractor for the 5th bit slows down the delta modulator because it must wait for the carry bit. The other four bits do not have to wait for a carry because they are formed in a 4 bit adder/subtractor IC which has an internal high speed carry look ahead circuit.

The complete circuit schematic for the high speed delta modulator is shown in Figure 17. The delta modulator contains 16 Schotky TLL integrated circuits, high speed D/A converters, and a high speed comparator.

When constructing a very high speed digital and analog system such as shown in Figure 17 it is necessary to adhere to certain construction practices or else "switching gliches", ringing, and oscillations may

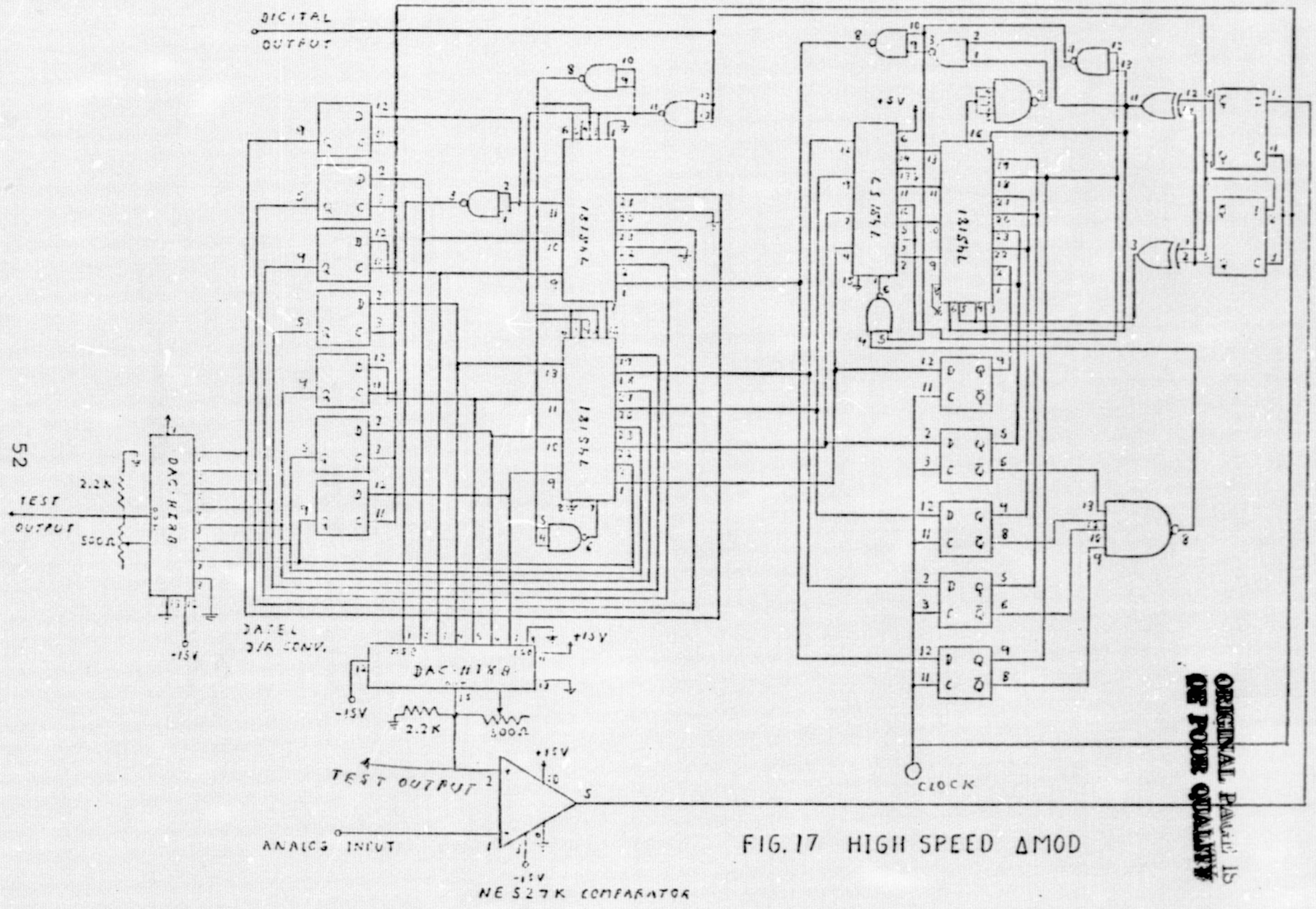


FIG. 17 HIGH SPEED ΔMOD

degrade the performance of the system. Some of the construction practices we used are listed below:

1) Bread board the unit on a double sided copper clad board. Use the lower copper side as a ground plane along which all wires are run. The upper copper side may be used as a power buss.
2) Put bypass capacitors between the power buss and the ground plane. It may be necessary to put bypass capacitors at every IC.
3) Isolate the analog portion of the system (D/A converter and comparator) from the switching noise of the digital section by providing separate bypass capacitors for all analog devices' power input pins.
4) High speed comparators love to oscillate as they pass through their linear region of operation. To prevent this, keep the input and output wires short, far apart, and if possible shielded.
5) Keep all wires as short as possible to prevent stray inductance and capacitance from causing ringing on the logic signals which might cause retriggering of the logic gates.

The high speed delta modulator was tested using the experimental setup shown in Figure 18. The setup functioned in the following manner: A scene from a slide was converted to a video signal by the TV camera. The video signal was band limited by an adjustable cutoff low pass filter whose output provided the input to the delta modulator. The delta

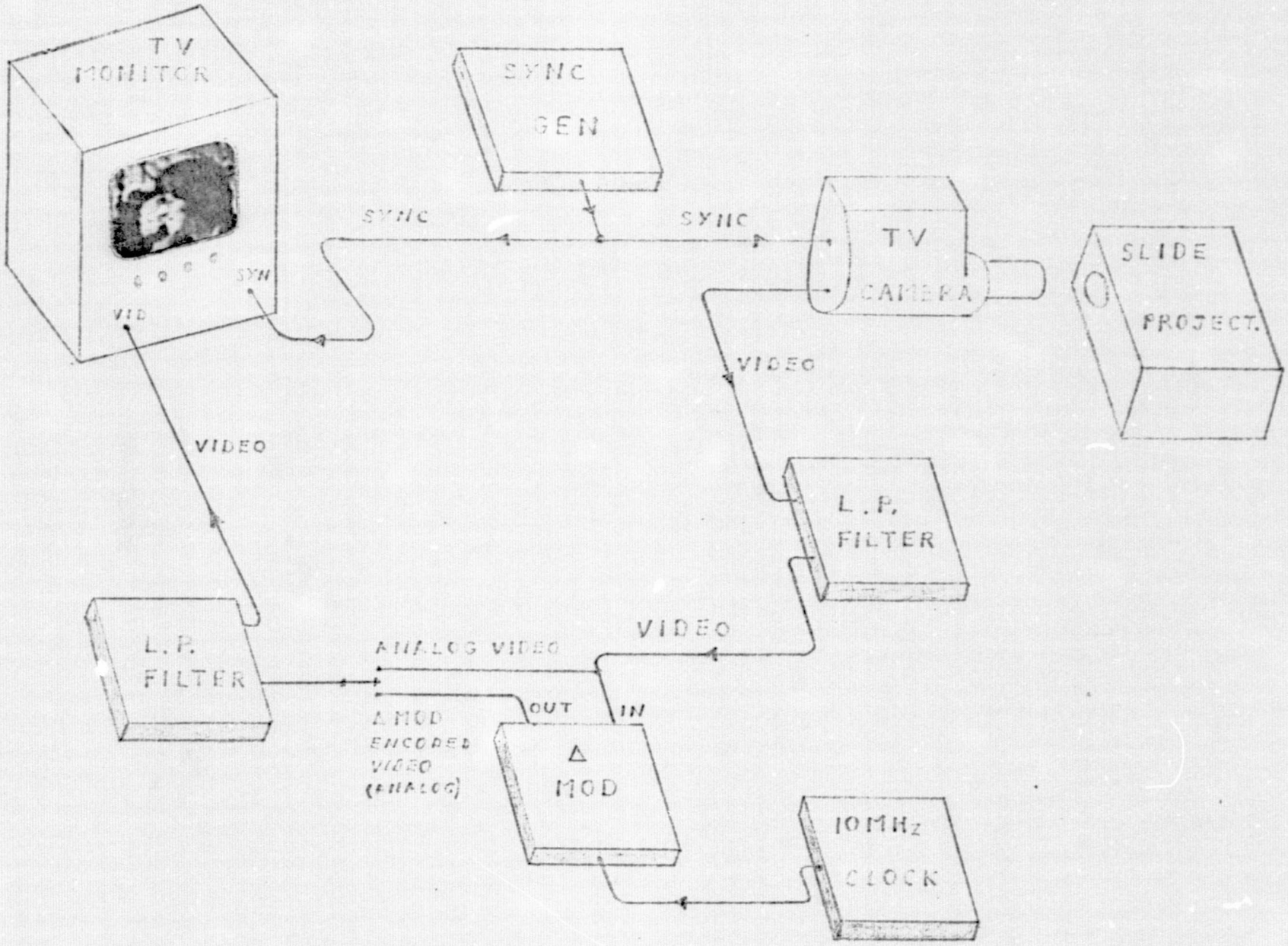


Fig.18 Real Time Video Processor For Delta Modulated Television Pictures

modulator was then clocked at its maximum rate of 10 MHz. The output of the delta modulator feeds another adjustable low pass filter whose cutoff is set to the same frequency as the previous filter. The processed picture is then displayed on a monitor and photographed. A switch is also provided so that a quick comparison can be made between the delta modulated encoded video signal and the unencoded video signal.

The output from the delta modulator used in Figure 18 was the test output marked on the diagrams of Figures 16 and 17. This output, in the absence of channel errors is the same as the output of the delta modulator receiver shown in Figure 10. This property made it unnecessary to build a receiver in order to test the delta modulator's ability to encode a video signal in the absence of channel errors.

The pictures in Figure 19 are representative of the quality of the pictures that we obtained from the test setup of Figure 18. The pictures on the left half of the page have been delta modulated while those on the right are the originals. The fact that the pictures on the right appear to be blurred is not due to bad photography, but is a consequence of low pass filtering the video signal. The bandwidth of the pictures, as determined by the settings on the filters of Figure 18, are given as follows: upper row, 2 MHz; middle row, 1.5 MHz; and lower row, 1 MHz.

Since the pictures in Figure 19 have frozen a single frame from the 30 frames per second TV camera and monitor, some effects of real time processing are not shown in the "frozen scenes" of Figure 19. I

will therefore try to describe these effects. In Figure 19, upper left hand corner, edge busyness can clearly be seen on the edges of vertical objects. The edge busyness appears as a slight horizontal displacement of each scanning line about the edge. Because some lines appear to be displaced more than others, straight vertical edges become wiggled. Now, imagine that successive frames of the same scene are flashed at the eye and each frame has its own wiggle pattern, as is the case with the real time processor, then the eye will see the edges flutter in time. This fluttering appears as a shimmering on the edges of objects. Although the shimmering is noticable it does not seem to cause a loss of resolution in the pictures and it is no more annoying than the edge busyness seen in Figure 19.

Just as the effect of viewing successive frames of a still picture cannot be seen in Figure 19, nor can the effect of the TV camera viewing a changing scene be determined from Figure 19. To determine the effect of the delta modulator on the video signal from a moving scene, we removed the slide projector from Figure 18 and aimed the camera at a room full of people at a party. The delta modulator showed no new kind of degradation for the moving scene. In fact, moving objects suffered slightly less edge busyness than still objects because the eye does not see moving objects very clearly.

If we assume that the pictures on the right in Figure 19 were encoded using 64 level PCM (sampling at the Nyquist rate) than we can calculate and compare the bit rate, and the number of bits per pel for

the pictures on the right with those on the left of Figure 19 from the following equations:

$$\text{Bits/Pel} = \frac{\text{Sampling Rate x Bits/Sample}}{\text{2 x Picture Bandwidth}}$$

$$\text{Bit Rate} = \text{Sampling Rate x Bits/Sample}$$

This yields: Table II

Picture	Delta Modulator		PCM	
	bit rate	bits per pixel	bit rate	bits per pixel
TOP	10 MHz	2.5	24 MHz	6
MIDDLE	10 MHz	3.3	18 MHz	6
BOTTOM	10 MHz	5.0	12 MHz	6

From the values in Table II, the pictures in Figure 19, as well as motion picture observations, we have concluded that our real time delta modulator can compress the bit rate over PCM by a factor of 2 with only a slight loss of detail. Although the loss of detail is minimal it is clear that the pictures in the left column are degraded over those in the right column of Figure 19. The degradation is most severe for the top row where the bit rate compression is 2.5 and almost unnoticable in the bottom row where the bit rate compression is 1.2.

For the most part, the real time delta modulator confirmed our

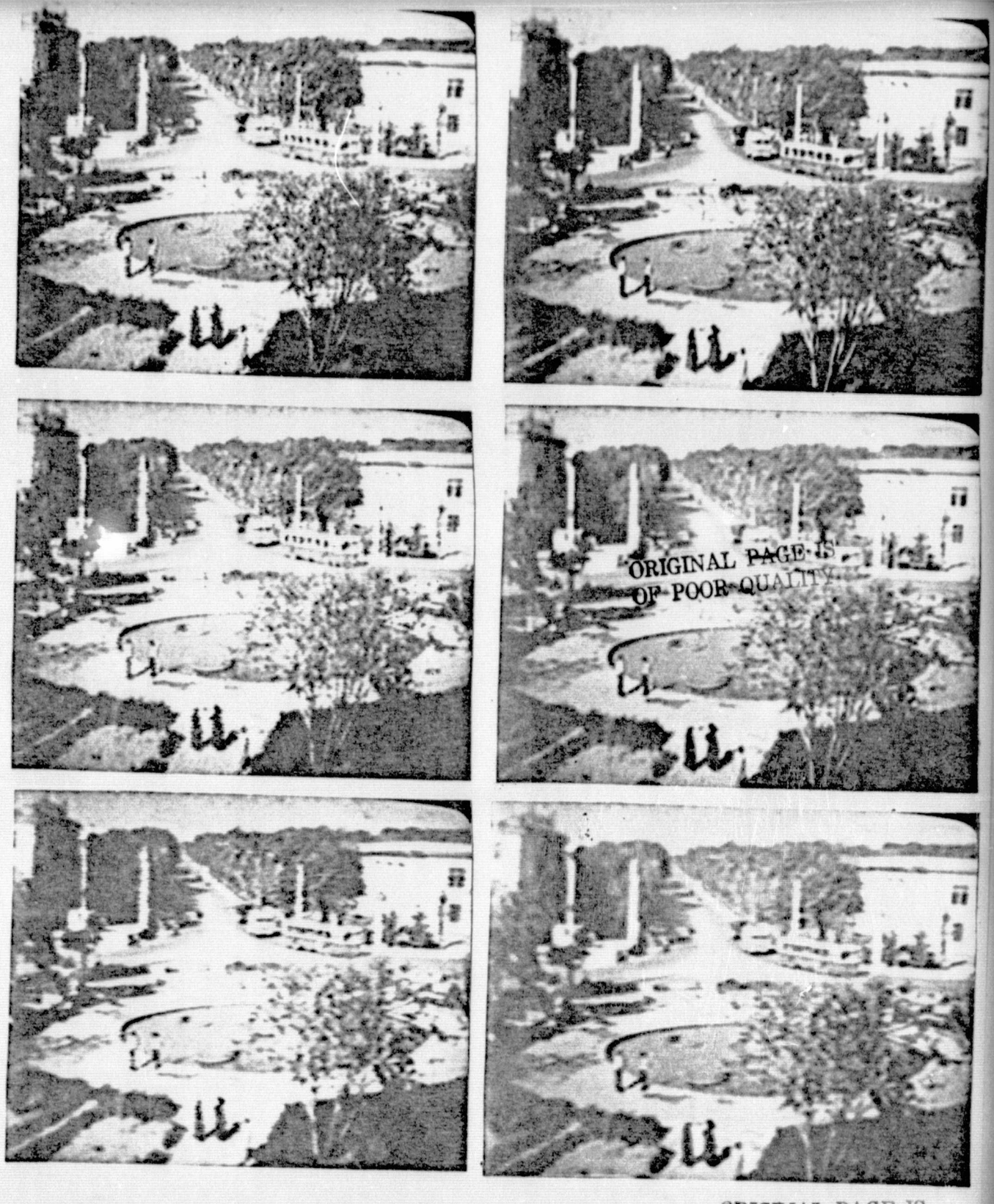


Fig. 19 A comparision of delta modulated pictures (left column) and unencoded pictures (right column) from a real time NTSC TV camera; Top row, $2MH_Z$ bandwidth; Middle row, $1.5MH_Z$ Bottom row, $1MH_Z$

computer simulations with just two exceptions. To reduce the granular noise (see Figure 12) to an imperceptible level and to provide the full range of gray levels seen in analog TV, 128 (not 64) quantization levels are needed. It was also determined that Y_{max} = 1/8 the p-p video signal gave the same quality picture as Y_{max} = 1/4 the p-p video signal.

CHAPTER IV

The Effect of Channel Errors on Delta Modulated Signals

When transmitting signals over a channel, channel errors will occur. For a binary channel, a channel error occurs when a +1 is sent and a -1 is received or a -1 is sent and a +1 is received. A communications channel designed to carry video information would be expected to have an error rate sufficiently low so that distortion due to channel errors would be barely perceptable in the received picture. This would be ground -30 db distortion.

It is desirable to achieve the -30 db distortion level with as little transmitter power as possible. The efficiency of a communications system is often judged by the amount of transmitted power it needs. We will judge the efficiency of the delta modulated channel against the PCM channel. But first we will investigate the effects of channel errors on the delta modulated video signal.

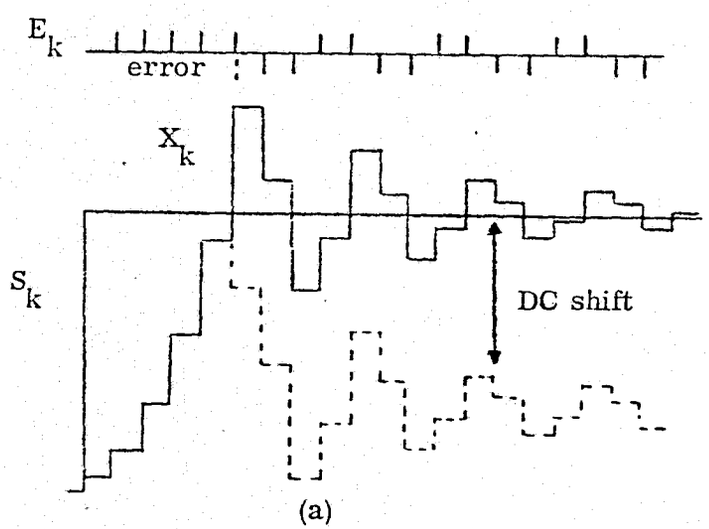
4.1 Qualitative Effects of Channel Errors

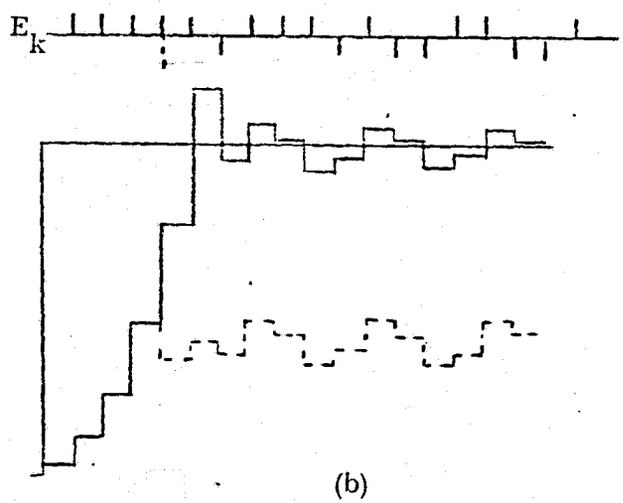
The response of the delta modulator decoder to a step input at the encoder in the presence of a single channel error is shown in Figure 20. The solid lines represent the output signal, X_k, of the decoder in the absence of channel errors, while the dashed lines represent the output signal of the decoder in the presence of a single channel error.

Channel errors have two effects on the delta modulator. The first and most obvious effect that can be seen in Figure 20 is that channel errors always cause a permanent, and usually large DC shift in the received signal. Less obvious is the fact that a single channel error may cause an increase, decrease or have no effect at all on the step size of the delta modulator encoder, and less obvious still, is the fact that step size errors, when they occur, become self correcting within a few samples. Because step size errors last for only a few samples, and delta modulators typically sample at several times the Nyquist rate of the input signal, the disturbance caused by step size errors is usually less than three pixels long and is completely masked by the disturbance caused by the large DC shift.

We have both quantitative results and experimental evidence to support the statements made in the preceding paragraph. The quantitative results were arrived at through the following process. Observe from Figure 20 and Equations 3.1.1, 3.1.2 and 3.1.3 that the step size, Y_k, will decrease while the delta modulator is tracking a constant DC level. If the constant signal level persists long enough the delta modulator reaches the minimum step size, Y_{min}, whether or not a channel error has occurred. When the delta modulator reaches the minimum step size, the step size error has corrected itself, since both the corrupted signal and the error free signal would have the same step size namely Y_{min}, the minimum step size.

The number of transmitted bits, N, that will reach the receiver





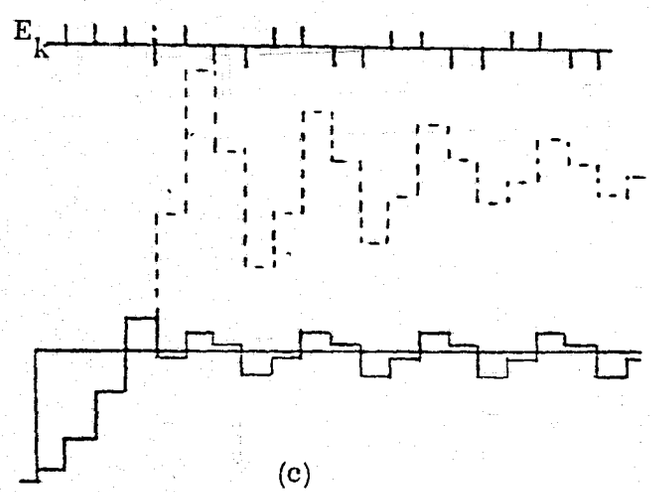


Fig.10 The effects of a single channel error on the output of the delta modulator
(a) No change in step size
(b) Decrease in step size
(c) Increase in step size

after the occurrence of a channel error, but before the step size corrects itself, has an upper bound for the delta modulator tracking a constant DC level. This upper bound can be shown to be

$$.75^{N/2} = Y_{min}/Y_{k+1}$$

$$N = \frac{2 \text{ In } Y_{min}/Y_{k+1}}{\text{In } .75} \qquad (4.1.1)$$

$N \equiv$ the number of transmitted bits until step size correction occurs
$Y_{k+1} \equiv$ the step size one sample time after the occurrence of the error
$Y_{min} \equiv$ the minimum allowable value for Y_k.

For the pictures in this paper the worst case parameters for Equation 6 are $Y_{min} = 1$ and $Y_{k+1} = 16$ which yield a worst case $N = 19$. If the delta modulator samples at 3 times the Nyquist rate of the video signal then the step size error should last for less than 6 pels if the conditions imposed in deriving equation 4.1.1 hold for a real picture.

The series of pictures in Figure 21 were taken to confirm the results of equation 4.1.1, i.e., step size errors quickly correct themselves, and to show that channel errors cause a large permanent shift in the DC level of the output of the delta modulator. Figure 21a shows the original picture without any errors. In Figure 21b channel errors were introduced at a rate of one error for every 2,500 transmitted bits. From Figure 21b we see that each channel error caused a shift in the

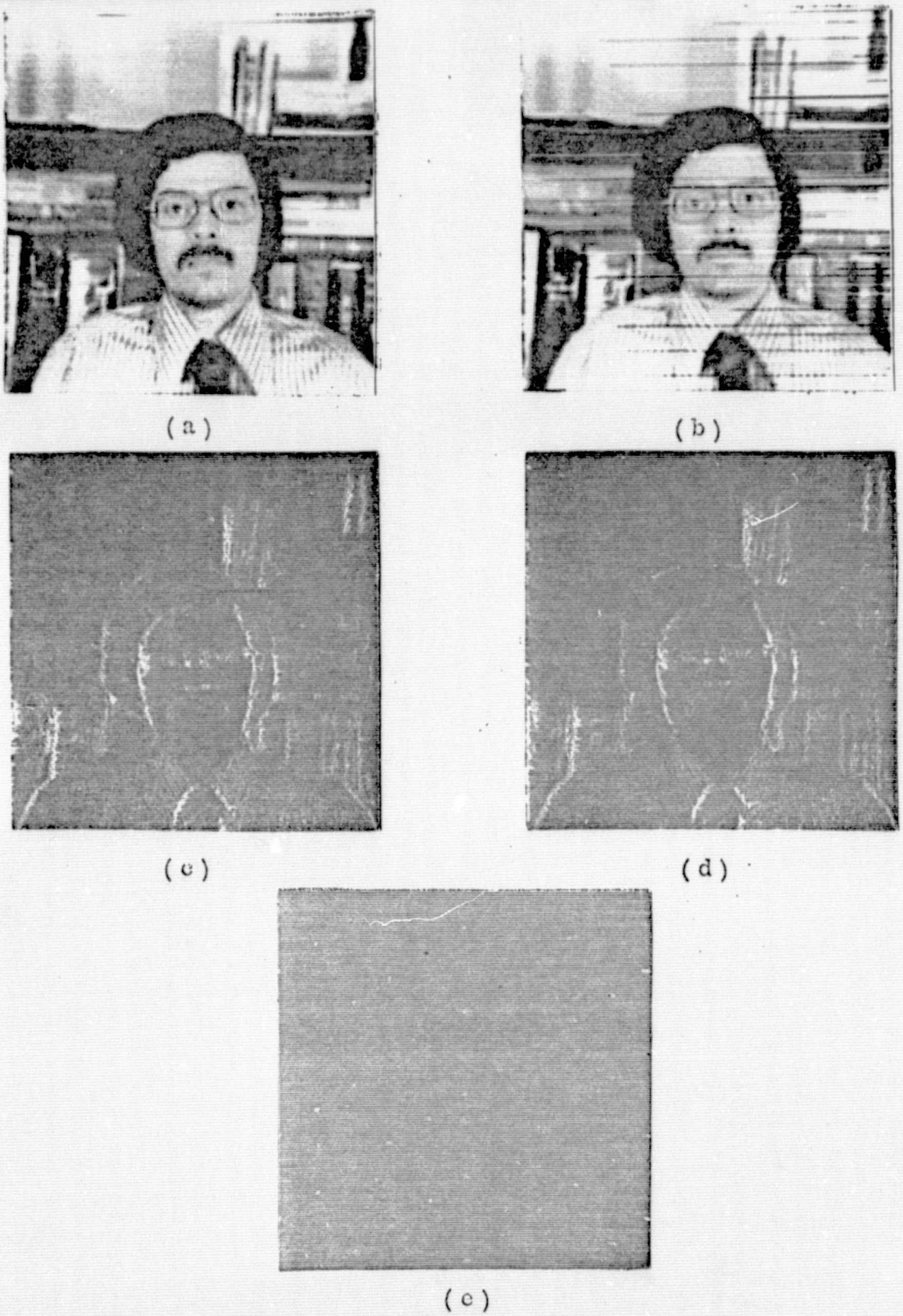


Fig. 21 The effects of chanuel errors on delta modulated encoded pictures. (a) Delta modalated picture without channel errors; 682 bits/line, 170 lines. (b) Error rate of 4×10^{-4}. (c) Aboslute value of Y_R (d) Absolute value of Y_R with channel errors (e) Difference between (c) and (d), i.e. step size errors.

level of the decoders output signal. This shift, seen as a streak, lasts until the end of the scanning line where the effect of the error is ended by resetting all the registers in the encoder and decoder to a fixed predetermined value. In Figure 21c we have displayed the absolute value of the step size without any channel errors and in Figure 21d we introduced channel errors. Figure 21e is the difference between Figure 21c and Figure 21d.

From Figure 21e we can see the effects of channel errors on the step size. The uniform gray background in Figure 21e represents zero difference between Figure 21c and Figure 21d or equivently, no step size error. The white and black dots are the regions where the difference between Figure 21c and Figure 21d was not zero or equivently, where a step size error exists. The short duration of the dots confirms the fact that step size errors become self correcting after a few samples.

The conclusions that we have drawn from our studies of the effects of channel errors on delta modulated video signals is that channel errors have a significant effect only on the DC level of the estimate, X_k, of the delta modulator decoder and that step size errors quickly correct themselves.

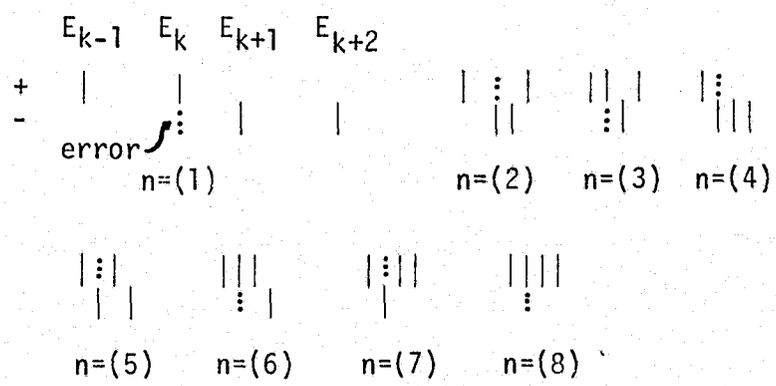
4.2 Quantitative Analysis of the Effects of Channel Errors

In section 3.2 the video delta modulator was compared with PCM with regards to encoding efficiency. In this section we will compare

the two techniques with regards to picture distortion caused by channel errors with respect to transmitter power.

For the delta modulator, channel errors cause distortion by causing a DC shift in the video signal. The DC shift becomes constant after the step size error corrects itself, and at the end of each scanning line the DC shift is also corrected (see Figure 8). The mean square value of the DC shift will be found by using the assumption that the DC shift is essentially constant within three transmitted E_k bits after the channel error has occurred for all E_k patterns. This is equivalent to assuming that the step size error corrects itself within three sample times.

There are eight 4 bit E_k patterns that produce no change in the step size. For these patterns the above assumption is exactly true. For the remaining eight patterns the assumption is almost true as attested to by Figure 21e. Eight of the 16 possible 4 bit E_k patterns are listed below. The other eight patterns are just their complement.



n is an index arbitrarily assigned to each pattern.

Let $|e_n|$ be the DC shift caused by a channel error in the nth E_k pattern. Then $|e_n|$ is given by:

$$|e_n| = |X - \hat{X}|$$

where X is the correct estimate and $\hat{X}$ is the incorrect estimate for E_k pattern "n".

Since the estimate is the sum of the step sizes, e_n is given by:

$$|e_n| = |\sum_{i=0}^{\infty} Y_i - \sum_{i=0}^{\infty} Y_i|$$

Y is equal to $\hat{Y}$ before the error at time k and it will be assumed equal to $\hat{Y}$ again by k+3; therefore,

$$|e_n| = |(Y_{k+1} + Y_{k+2} + Y_{k+3}) - (\hat{Y}_{k+1} + \hat{Y}_{k+2} + \hat{Y}_{k+3})| \quad (4.2.1)$$

To be consistent with equation 3.1.3, we choose an error in E_k at time k to cause the first step size error to appear in Y_{k+1}.

By substituting equation 3.1.3 into 4.2.1 we can find $|e_n(Y_k)|$, the DC shift as a function of Y_k. Equation 3.1.3 has three parts. The part to be substituted into equation 4.2.1 depends on the step size. If the magnitude of all the step sizes in equation 4.2.1 are between $2Y_{min}$ and Y_{max} then equation 3.1.3a is used to find $e_n(Y_k)$. Using equation 3.1.3a to find $e_n(Y_k)$ and letting $\hat{E}_k$ be the incorrect

E_k we have:

$$e_n(Y_k) = |Y_k|(E_k + \tfrac{1}{2}E_{k-1})$$
$$+ ||Y_k|(E_k + \tfrac{1}{2}E_{k-1})|(E_{k+1} + \tfrac{1}{2}E_k)$$
$$+ |||Y_k|(E_k + \tfrac{1}{2}E_{k-1})|(E_{k+1} + \tfrac{1}{2}E_k)|(E_{k+2} + \tfrac{1}{2}E_{k+1})$$
$$- |Y_k|(\hat{E}_k + \tfrac{1}{2}E_{k-1}) - ||Y_k|(\hat{E}_k + \tfrac{1}{2}E_{k-1})|(E_{k+1} + \tfrac{1}{2}\hat{E}_k)$$
$$- |||Y_k|(\hat{E}_k + \tfrac{1}{2}E_{k-1})|(E_{k+1} + \tfrac{1}{2}\hat{E}_k)|(E_{k+2} + \tfrac{1}{2}E_{k+1}) \qquad (4.2.2)$$

Because E_k is ± 1 we can use the identity

$$\left||Y_k|(E_k + \tfrac{1}{2}E_{k-1})\right| \equiv E_k|Y_k|(E_k + \tfrac{1}{2}E_{k-1})$$

to simplify equation 4.2.2. Doing so yields:

$$\boxed{|e_n(Y_k)| = |Y_k(3.5 + E_{k+1}E_{k+2} + E_{k-1}E_{k+2} + 1.5\,E_{k-1}E_{k+1})|} \qquad (4.2.2a)$$

Equation 4.2.2a is accurate for values of $|Y_k|$ between $2Y_{min}$ and $Y_{max}/1.5$. For values outside this range equation 4.2.1 must be evaluated for each value of n using all three parts of equation 3.1.3.

Evaluating equation 4.2.1 for values of $|Y_k|$ that lay outside the range of equation 4.2.2a with $Y_{min} = 1$ and $Y_{max} = 15$ yields:

FOR $Y_k = 1$

$$|e_1(1)| = |(2 - 1 - 2) - (- 2 - 3 - 4)| = 8$$
$$|e_2(1)| = |(- 2 - 3 + 1) - (2 - 1 + 3)| = 7$$
$$|e_3(1)| = |(2 - 1 + 2) - (- 2 - 3 + 1)| = 7$$
$$|e_4(1)| = |(- 2 - 3 - 4) - (2 - 1 - 2)| = 8$$

FOR $Y_k = 9$

$e_1(9), e_2(9), e_3(9), e_4(9), e_5(9), e_6(9)$ Eq. 4.2.2a is valid

$$|e_7(9)| = |e_8(9)| = |(- 4 + 2 + 3) - (13 + 15 + 15)| = 42$$

FOR $Y_k = 10$

$e_1(10), e_2(10), e_3(10), e_4(10)$ Eq. 4.2.2a is valid

$$|e_5(10)| = |e_6(10)| = |(- 5 + 2 - 1) - (15 + 15 - 7)| = 27$$
$$|e_7(10)| = |e_8(10)| = |(- 5 + 2 + 3) - (15 + 15 + 15)| = 45$$

FOR $Y_k = 13$

($Y_k = 11$, 12 and 14 does not exist)

$e_1(13), e_2(13), e_3(13), e_4(13)$ Eq. 4.2.2a is valid

$$|e_5(13)| = |e_6(13)| = |(- 6 + 3 - 1) - (15 + 15 - 7)| = 27$$
$$|e_7(13)| = |e_8(13)| = |(- 6 + 3 + 4) - (15 + 15 + 15)| = 44$$

FOR Y = 15

$e_1(15)$, $e_2(15)$, $e_3(15)$, $e_4(15)$ Eq. 4.2.2a is valid

$|e_5(15)| = |e_6(15)| = |(-7 + 3 - 1) - (15 + 15 - 7)| = 28$

$|e_7(15)| = |e_8(15)| = |(-7 + 3 + 4) - (15 + 15 + 15)| = 45$

From the above results and equation 4.2.2a, $e_n(Y_k)$ for the eight E_k patterns is found to be

$$e_1(Y_k) = e_2(Y_k) = e_3(Y_k) = e_4(Y_k) = \begin{cases} 2Y_k & 1 < Y_k \leq 15 \\ \approx 7.5 & Y_k = 1 \end{cases} \qquad (4.2.3)$$

$$e_5(Y_k) = e_6(Y_k) = \begin{cases} 3Y_k & 1 < Y_k \leq 9 \\ 7 & Y_k = 1 \\ 27 & Y_k \geq 9 \end{cases} \qquad (4.2.4)$$

$$e_7(Y_k) = e_8(Y_k) = \begin{cases} 6.5Y_k & 1 < Y_k \leq 8 \\ 8 & Y_k = 1 \\ \approx 43 & Y_k \geq 9 \end{cases} \qquad (4.2.5)$$

The mean square value of any one of the DC shifts, $e_n(Y_k)$, $n = 1, 2, \ldots 8$ is given by:

$$\overline{e_n^2} = \sum_{Y_k=0}^{Y_{max}} [e_n(Y_k)]^2 \, p(Y_k/n) \qquad (4.2.6)$$

$P(Y_k/n)$ = Probability of Y_k given that E_k pattern n will occur.

The average mean square value of all the DC shifts is given by

$$\overline{e^2} = \sum_{n=1}^{8} \overline{e_n^2}\, p_n(n) \tag{4.2.7}$$

$P_n(n)$ = Probability of E_k pattern n plus the compliment of n.

Assuming that $p_{n,y}(Y_k/n) = p(Y_k)$ and combining equations 4.2.3 to 4.2.7 the expression for $\overline{e^2}$ becomes:

$$\begin{aligned}\overline{e^2} = {} & \left[7.5^2 p_y(1) + \sum_{Y_k=2}^{15} (2Y_k)^2\, p(Y_k)\right]\left[p_n(1) + p_n(2) + p_n(3) + p_n(4)\right] \\ & + \left[7^2 p_y(1) + \sum_{Y_k=9}^{15} 27^2 p(Y_k) + \sum_{Y_k=2}^{8} (3Y_k)^2 p(Y_k)\right]\left[p_n(5) + p_n(6)\right] \\ & + \left[8^2 p_y(1) + \sum_{Y_k=9}^{15} 43^2 p(Y_k) + \sum_{Y_k=2}^{8} (6.5Y_k)^2 p(Y_k)\right]\left[p_n(7) + p_n(8)\right]\end{aligned} \tag{4.2.8}$$

Define $\alpha_{1-4} = p_n(1) + p_n(2) + p_n(3) + p_n(4)$

$\alpha_{5-6} = p_n(5) + p_n(6)$

$\alpha_{7-8} = p_n(7) + p_n(8)$

Then we can rewrite equation 4.2.8 as

$$\overline{e^2} = (4\alpha_{1-4} + 9\alpha_{5-6} + 42\alpha_{7-8}) \sum_{Y_k=2}^{8} Y_k^2 p(Y_k)$$
$$+ (56\alpha_{1-4} + 49\alpha_{5-6} + 64\alpha_{7-8})\, p_y(1)$$
$$+ (729\alpha_{5-6} + 1850\alpha_{7-8}) \sum_{Y_k=9}^{15} p(Y_k)$$
$$+ 4\alpha_{1-4} \sum_{Y_k=9}^{15} Y_k^2 p(Y_k) \qquad (4.2.8a)$$

The mean square value of the DC shift, $\overline{e^2}$, caused by a channel error can be calculated from equation 4.2.8a if the step size statistics and the statistics of the E_k patterns are known. The real time delta modulator was set up with special counters so that these statistics could be measured. The sampling rate was three times the Nyquist rate and the pictures used are shown in Figure 22. The results are given in Table III.

From Table III we can see that all eight E_k patterns are equally probable to within a factor of 2.5. The range of probabilities on the step sizes, however, is nearly 100 to 1. Because the step size logic always truncates the step size to the nearest, smallest integer the step size values 8, 11, 12 and 14 never occur. In general the probability of the step size decreases monotonically with increasing step size. Exception to the monotonicity such as step size 5, results because of the way the step size logic truncates. Step size 5 can be

reached only by halving step size 10; therefore, step size 5 must be less probable than step size 10.

From Table III and equation 4.2.8a $\overline{e^2}$ is calculated to be

$$\boxed{\overline{e^2} = 146} \qquad \text{BOY} \qquad (4.2.9)$$

Knowing the value of $\overline{e^2}$, we can determine the transmitter power needed so that distortion caused by channel errors would be at least -30 db below the picture power. For the determination of transmitter power we assumed the presence of white gausian channel noise and antipodal signals. The transmitter power was determined as follows:

D_e = picture distortion energy/error
N_p = picture distortion energy/pel
N = number of transmitted bits/pel
L = number of pels/line
Pe = probability of channel error/transmitted bit

When an error occurs it lasts until the end of the scanning line. Since it is equally likely for an error to occur anywhere along the line, on the average an error will last for one-half of a line. Therefore, the picture distortion energy/error is

$$D_e = \frac{L}{2}\,\overline{e^2} \qquad (4.2.10)$$

N_p is the distortion energy per pel averaged over all the pels in the picture and is given by

$$N_p = D_e PeN$$

Substituting from equation 4.2.10

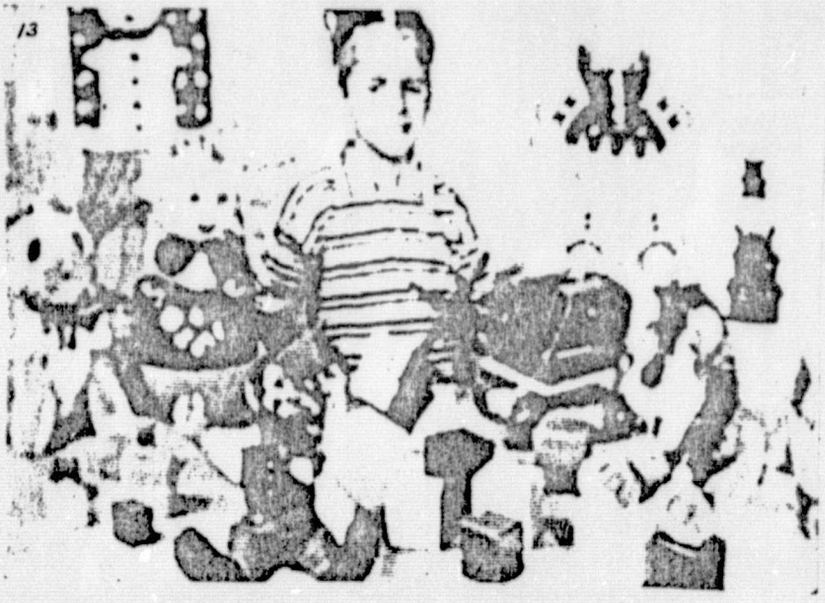
$$N_p = \frac{L\overline{e^2}PeN}{2} \equiv \text{distortion energy/pel} \qquad (4.2.11)$$

TABLE III

BOY					GIRL				
$\|Y_k\|$	$P(\|Y_k\|)$	n	E_k's	P(n)	$\|Y_k\|$	$P(\|Y_k\|)$	n	E_k's	P(n)
1	.30	1	1100	.21	1	.30	1	1100	.24
2	.35	2	1001	.15	2	.35	2	1001	.17
3	.20	3	1101	.09	3	.18	3	1101	.078
4	.081	4	1000	.14	4	.072	4	1000	.15
5	.0016	5	1010	.08	5	.0016	5	1010	.063
6	.037	6	1110	.15	6	.037	6	1110	.15
7	.0043	7	1011	.09	7	.004	7	1011	.08
8	.0	8	1111	.09	8	.0	8	1111	.07
9	.016				9	.017			
10	.002				10	.0021			
11	.0				11	.0			
12	.0				12	.0			
13	.0065				13	.0073			
14	.0				14	.0			
15	.0054				15	.0052			

Note: Pn(n) includes the pattern n plus its complement.

Step size statistics and E_k statistics for the Boy and Girl of Figure 20 sampled by the delta modulator at 3 times the Nyquist rate.



BOY

GIRL

Fig. 22 Pictures used for delta modulator's statistics

Defining:

$S_p \equiv$ picture energy/pel
$E_b \equiv$ energy/bit
$N_o \equiv$ power spectral density of channel noise
$R_s \equiv$ number of transmitted bits/sec
$S_T \equiv$ transmitter power

If we assume that it is equally likely for each pel of the picture to take on any one of q quantization levels and that the quantization levels range from 0 to q then S_p is given by:

$$S_p = \frac{1}{q}\sum_{x=0}^{q} x^2 \simeq \frac{1}{q}\int_0^q x^2 dx$$

$$S_p = \frac{q^2}{3} \equiv \text{picture energy/pel} \qquad (4.2.12)$$

From equation 4.2.11 and 4.2.12 we can find S_p/N_p and then we can find Pe by letting the picture signal to noise ratio, S_p/N_p, equal +30 db

$$\frac{S_p}{N_p} = \frac{q^2/3}{L\overline{e^2}PeN/2} = 1000 \qquad (4.2.13)$$

Letting $q = 64$ quantization levels
$N = 3$ transmitted bits/pel
$L = 500$ pels/line
$\overline{e^2} = 146$

Then from equation 4.2.13

$$Pe = 1.2 \times 10^{-5} \tag{4.2.14}$$

The transmitter power can be found by relating Pe to E_b/n_o and then E_b/n_o to S_T. The energy per bit, E_b, is related to Pe by the following well known function:

$$Pe = Q(\sqrt{2E_b/n_o})$$

which is plotted in Figure 23 . From Figure 23 we see that for $Pe = 1.2 \times 10^{-5}$

$$E_b/n_o = 9.5 \text{ db} \tag{4.2.15}$$

The transmitter power is related to E_b through the bit rate, R_s by:

$$S_T = E_b R_s \tag{4.2.16}$$

R_s is given by the Nyquist sampling theorem as:

$$R_s = N(2fm) \tag{4.2.17}$$

where fm is the picture bandwidth and N is the number of transmitted bits/pel. Substituting equation 4.2.17 into 4.2.16 yields:

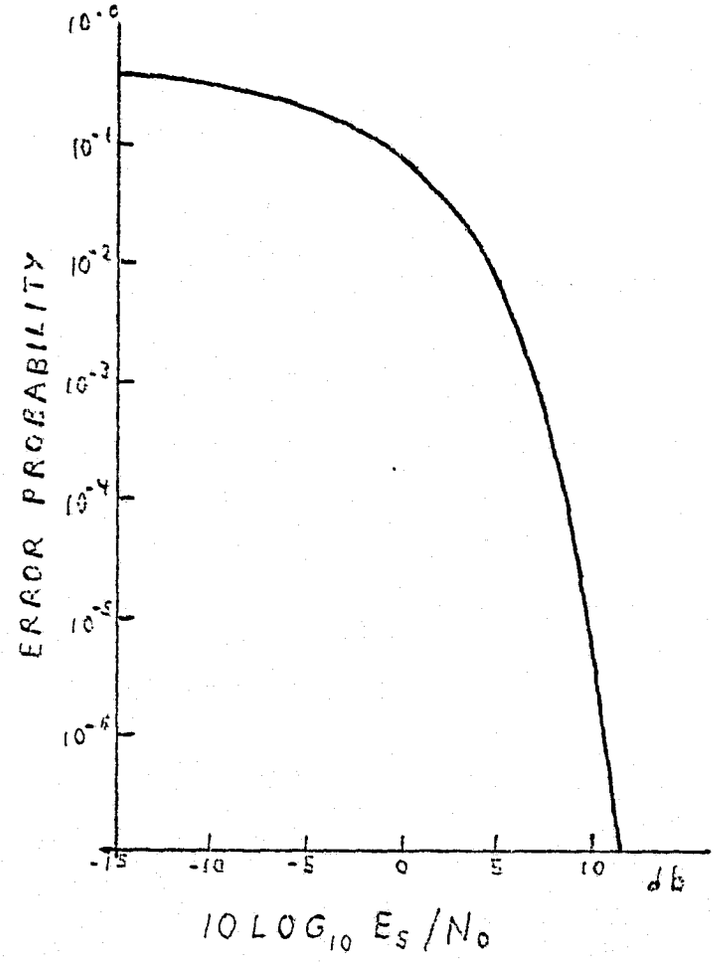


FIG 23 PROBABILITY OF ERROR FOR BINARY ANTIPODAL SIGNALS

$$S_T = E_b N(2fm) \tag{2.4.18}$$

Substituting equation 4.2.15 into 4.2.18 and letting N=3 yields:

$$\boxed{\frac{S_T}{n_o fm} = 17.4 \text{ db}} \tag{4.2.19}$$

Equation 4.2.19 gives us the transmitter power needed for a delta modulated picture to bring the picture distortion due to channel errors down to -30 db when the delta modulator samples at 3 bits/pel.

The transmitted power, S_T, required by a PCM encoded channel to achieve -30 db distortion due to channel errors for antipodal signaling in the presence of white gausian channel noise can be found as follows:

An error may occur with equal likelihood in any of the N bits in a pel, and if an error in the least significant bit causes 1 quantization level of distortion and an error in the Nth bit causes 2^{N-1} quantization levels of distortion then:

$$D_e = \frac{1}{N}(1 + 2^2 + 4^2 \ldots + (2^{N-1})^2)$$

$$D_e \approx \frac{2^{2N}}{3N} \tag{4.2.20}$$

Also

$$Np = (De)(Pe)(N) \tag{4.2.21}$$

Substituting 4.2.20 into 4.2.21 yields:

$$Np = \frac{2^{2N}Pe}{3} \tag{4.2.22}$$

From equation 4.2.12 the signal power is:

$$Sp = \frac{2^{2N}}{3} \tag{4.2.23}$$

since $q^2 = 2^{2N}$ for PCM.

From 4.2.22 and 4.2.23 we can find Sp/Np in terms of Pe and N and then we can solve for Pe by letting the picture signal to noise ratio, Sp/Np, equal +30 db.

$$\frac{Sp}{Np} = \frac{2^{2N}/3}{2^{2N}Pe/3} = +30 \text{ db}$$

$$\frac{Sp}{Np} = \frac{1}{Pe} = 1000 \tag{4.2.24}$$

$$Pe = 10^{-3} \text{ errors/bit} \tag{4.2.25}$$

Equation 4.2.24 states that the picture noise to signal ratio from channel errors is equal to the probability of error for a PCM transmission system.

The transmitter power can be found by relating Pe to E_b and then E_b to S_T as was done for the delta modulated channel.

The energy per bit, E_b, is related to Pe by:

$$Pe = Q(\sqrt{2E_b/n_o}) \tag{4.2.26}$$

From Figure 23 we see that for $Pe = 10^{-3}$

$$E_b/n_o = 7 \text{ db} \tag{4.2.27}$$

From equation 4.2.27 and 4.2.18

$$\frac{S_T}{n_o fm} = 7 \text{ db} + 10 \log 2N$$

A typical value for N for PCM is 6 bits/pel. Therefore,

$$\boxed{\frac{S_T}{n_o fm} = 18 \text{ db}} \tag{4.2.28}$$

Equation 4.2.28 gives us the transmitter power for PCM needed to bring the picture distortion due to channel errors down to -30 db.

4.3 Conclusions

In summarizing the results of section 4.2 we can make the following observations:

1. The power of the distortion in a PCM channel caused by a single channel error as given by equation 4.2.20 is similar to the power of the distortion in the delta modulated channel as given by equation 4.2.9.

2. The energy per error for the delta modulated channel is over 200 times greater than for the PCM channel because the delta modulator's error lasts for 250 pels on the average.

3. For a -30 db picture distortion level from channel errors, the PCM channel requires a 10^{-3} bit error rate whereas the delta modulator requires a bit error rate as low as 10^{-5}.

4. From equation 4.2.19 and 4.2.28 we see that both PCM and delta modulation require the same transmitter power to achieve -30 db distortion due to channel errors. This is because the delta modulator transmits at one-half the PCM rate and can thus achieve a probability of error more than two orders of magnitude lower than the PCM channel with the same transmitter power as PCM.

CHAPTER V

Error Correcting Algorithms

By incorporating error correcting algorithms into the delta modulator design it is possible to reduce transmitter power and still maintain a picture signal to distortion ratio of 30 db. In this chapter, three error correcting algorithms are explored.

5.1 Direct Approach

It is possible to correct the DC level of the decoders estimate by periodically sending the transmitters current estimate, X_k, to the receiver. The effect of this correction technique, shown in Figure 24, is to shorten the length of the error streaks. The more often the transmitters estimate is sent to the receiver to correct the estimate at the receiver the shorter the streaks become. Unfortunately the more often we send the transmitters estimate, the more we must increase the transmission rate to accomodate this extra information.

The equation that relates the several parameters that determine the increase in transmission rate is given by equation 5.1.1.

$$R'_s = R_s(1 + \frac{cb}{s}) \qquad (5.1.1)$$

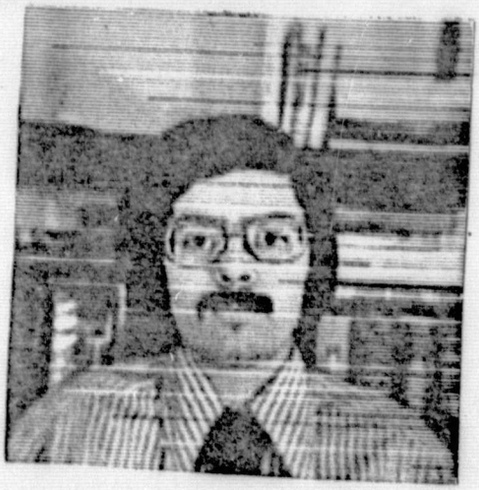
where

$R'_s \equiv$ The bit rate with correction
$R_s \equiv$ The bit rate without correction
$c \equiv$ The number of times X_k is sent to the receiver per line
$b \equiv$ The number of bits in each X_k sent to the receiver
$s \equiv$ The number of E_k's transmitted per line

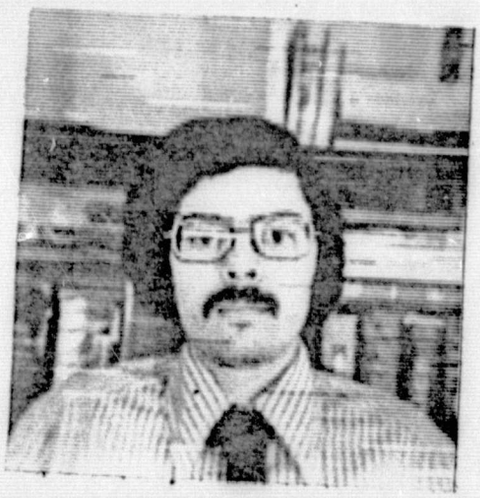
From equation 5.1.1 it is apparent that "c" and "b" should be as small as possible. "c" is lower bounded by the desired degree of correction required in the picture since the length of the remaining error streaks in the received picture are inversely proportional to c, and b is lower bounded by the accuracy of the correction.

To understand the effect of "b" on the correction algorithm refer to Figure 25. Figure 25 shows how the correction algorithm is implemented. First the "b" most significant bits of the transmitter's estimate are sent to the receiver via a PCM format. Upon receiving the PCM word the receiver sets the "b" most significant bits in its estimate equal to the transmitter's estimate. Then both the transmitter and receiver set their b + 1 most significant bit to 1 and all less significant bits to zero. The total effect is to make the transmitter's and receiver's estimate equal, and to introduce an inaccuracy in the estimate of maximum value equal to 2^{-b-1} times the maximum amplitude of the picture.

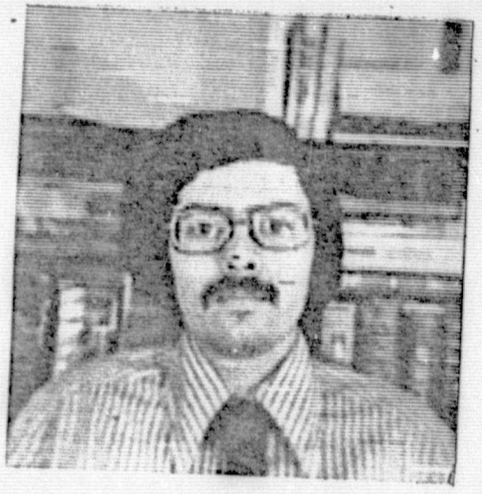
The inaccuracy in the estimates introduced by the correction algorithm is shown in Figure 26. The shaded area is the error introduced by the correction algorithm and is equal to the difference between the transmitter's estimate without the correction algorithm (solid



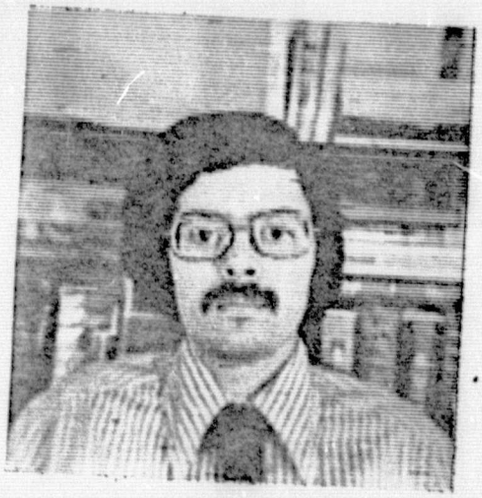
(a) Nosiy picture; 3×10^{-4} error rate; 1024 bits/line; 170 lines.



(b) 4 corrections per line; 1.5% increase in bit rate.



(c) 8 corrections per line; 3% increase in bit rate.



(d) 16 corrections per line; 6% increase in bit rate.

Fig.24 The effect of sending the encoders estimate to the decoder to acheive error correction.

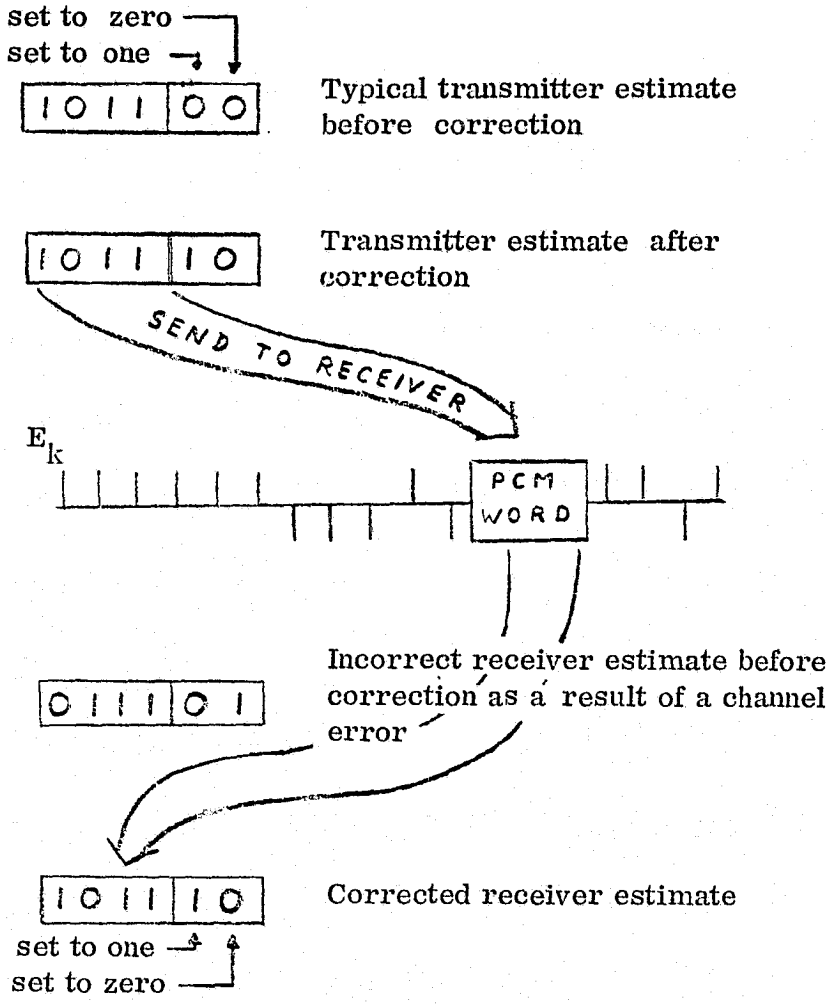


Fig. 25 Error correction is achieved by sending the "b" most significant bits of the transmitter's estimate to the receiver.

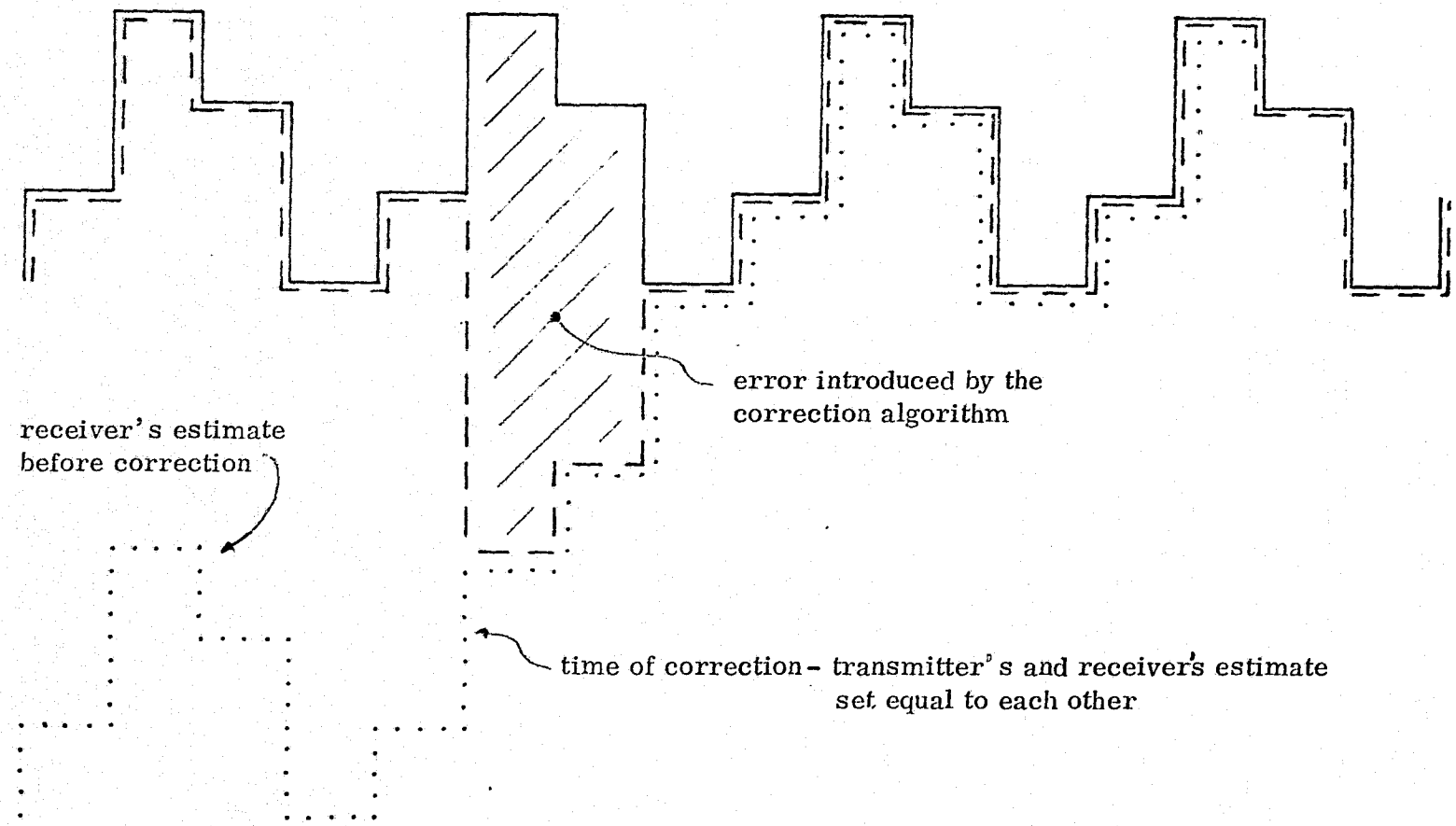


Fig. 26 (a) The solid line shows the correct estimate in the minimum step size pattern
(b) The dashed line shows the same situation but with the correction
(c) The dotted line shows the receiver's estimate before correction

line) and the transmitter's estimate with correction (dashed line). Notice also how the correction algorithm eliminated the DC shift in the receiver's estimate (dotted line) and thus corrected for a channel error.

Figure 27 shows the effect of using different values for "b" in the correction algorithm. In Figure 27a the maximum value of the error caused by the correction algorithm is 25% of the peak to peak video signal. Such a large error produces objectionable black bars as seen from Figure 27a. For b = 4, the error is only 3.1% and is not noticeable as attested to by Figure 27d.

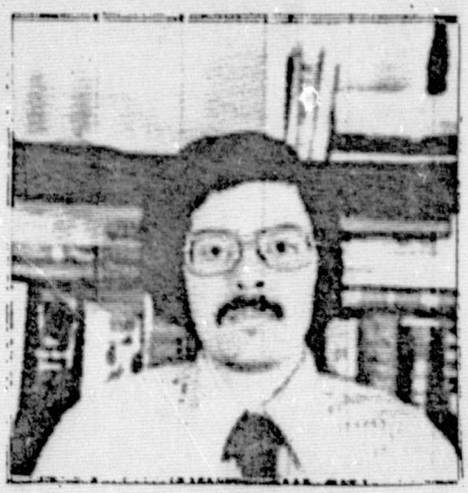
In section 4.2 the transmitter power needed to achieve a -30 db distortion level without error correction was found. In this section we will calculate the transmitter power required to achieve a -30 db distortion level with error correction.

Proceeding as in section 4.2 we have

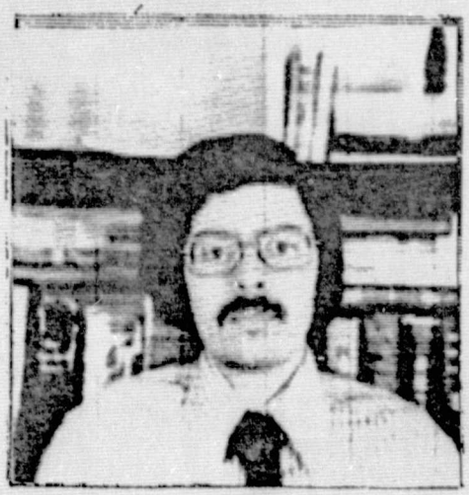
$$D_e = \frac{L}{2(c+1)} \overline{e^2} \equiv \text{distortion energy/error} \qquad (5.1.2)$$

where L/2(c+1) is the average length of an error streak with "c" corrections per line.

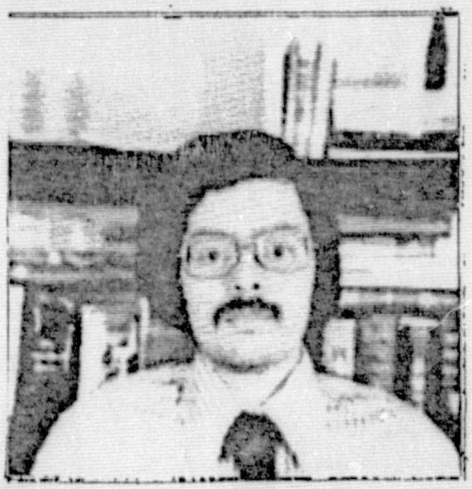
The error correction algorithm increases the average number of bits per pel given as N in section 4.2 by the amount bc/L. Let



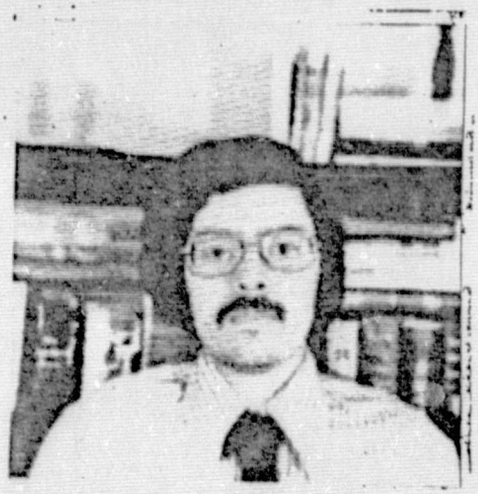
(a) b = 1



(b) b = 2



(c) b = 3



(d) b = 4

Fig. 27 The effect of sending only the "b" most significant bits of the encoders estimate to the decoder for error correction.

$N_c = N + bc/L$ be the number of bits per pel with correction, then from section 4.2 and equation 5.1.2

$$Np = D_e Pe N_c$$

$$Np = \frac{\overline{Le^2}\, Pe(N + bc/L)}{2(c+1)} = \text{distortion energy/pel} \qquad (5.1.3)$$

Equation 5.1.3 has assumed that an error in the correction bits has the same effect as an error in an E_k bit.

From equation 4.2.12 and 5.1.3 and the fact that Sp/Np is set equal to 30 db we have

$$\frac{Sp}{Np} = \frac{q^2/3}{\overline{Le^2}\, Pe\,(N + bc/L)/2(c + 1)} = 1000$$

Solving for Pe

$$Pe = \frac{2(c + 1)q^2}{3\overline{Le^2}\, 10^{+3}\, (N + \frac{bc}{L})} \qquad (5.1.4)$$

Using the parameter values of section 4.2 q = 64, N = 3, L = 500, $\overline{e^2} = 146$, and b = 4, Pe is found to be

$$Pe = 1.2 \times 10^{-5} \left(\frac{c + 1}{1 + .0027c} \right) \qquad (5.1.5)$$

Pe is the probability of error as a function of the number of corrections per line required to obtain a picture signal to distortion ratio (due to channel errors) of 30 db.

Similarly from equation 4.2.18 and $Nc = N + bc/L$, S_T is given by

$$S_T = E_b N_c (2fm)$$

$$\frac{S_T}{n_o fm} = \frac{E_b}{n_o} (6 + .016c) \qquad (5.1.6)$$

Figure 28 is a plot of $S_T/n_o fm$ vs c for b = 4 and 6. The plot was constructed by choosing a value for "c", then solving equation 5.1.5 for Pe, and then using Figure 21 to obtain E_b/n_o for that value of Pe. Equation 5.1.6 was then used to calculate $S_T/n_o fm$ from E_b/n_o and c.

From Figure 28 we can see that the optimum number of corrections per line is about 50 and the savings in transmitted power is about 1.6 db. The price we would pay in using the this scheme to save 1.6 db of transmitter power is a 16% increase in bandwidth and the addition of timing and synchronizing circuiting at the receiver needed to separate the correction bits from the delta modulator bits.

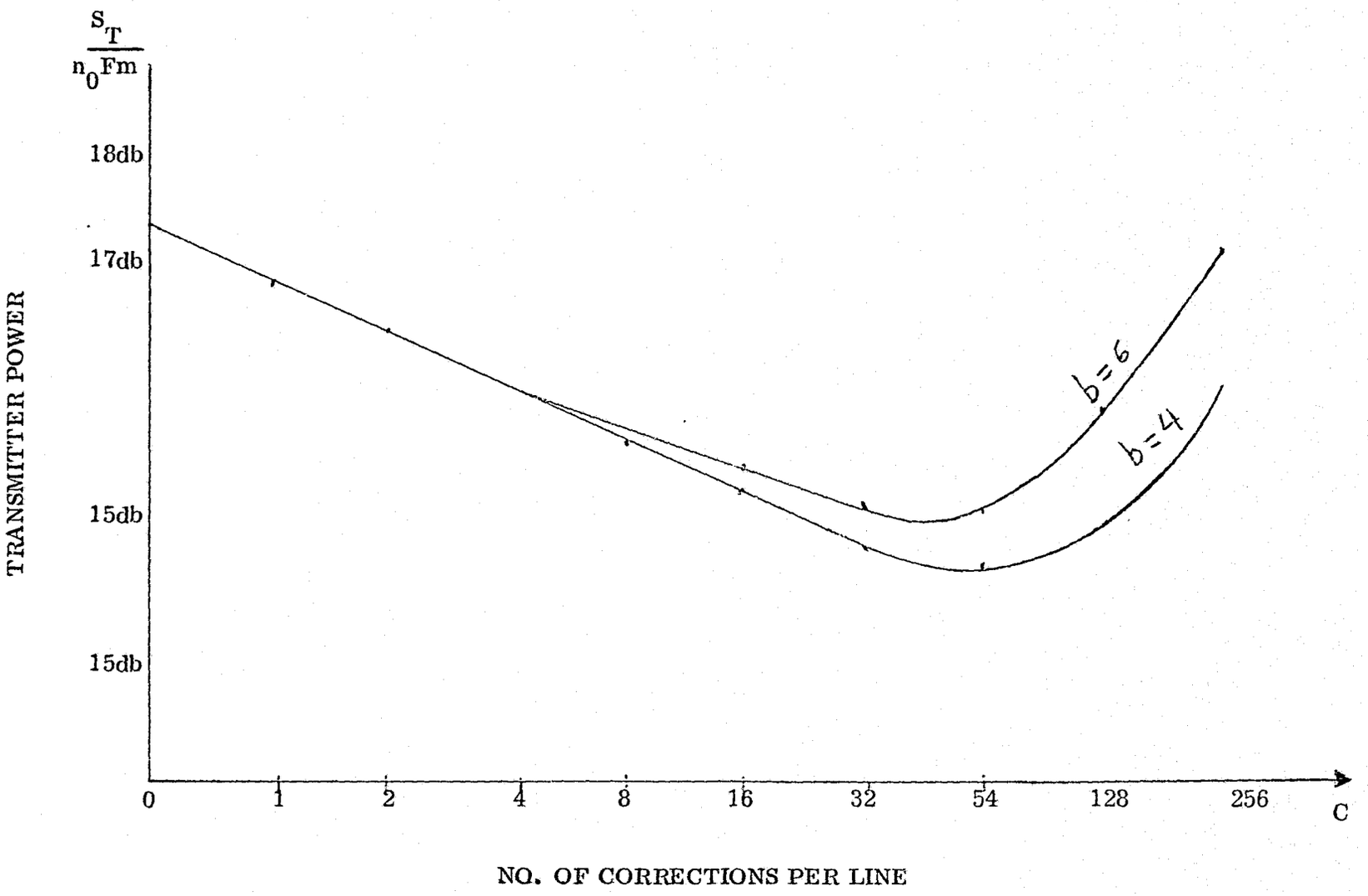


Fig. 28 The number of corrections per line vs. the transmitted power required to maintain a picture signal to distortion ratio of +30db due to channel errors

5.2 Leaky Integrator

The effects of channel errors on delta modulated pictures can be reduced by introducing leaky integrators to the feed back loop of the delta modulator encoder and decoder. Leaky integration is achieved by introducing the factor α, ($\alpha < 1$), into equation 3.1.1 as shown in equation 5.2.1

$$X_{k+1} = \alpha X_k + Y_{k+1} \qquad (5.2.1)$$

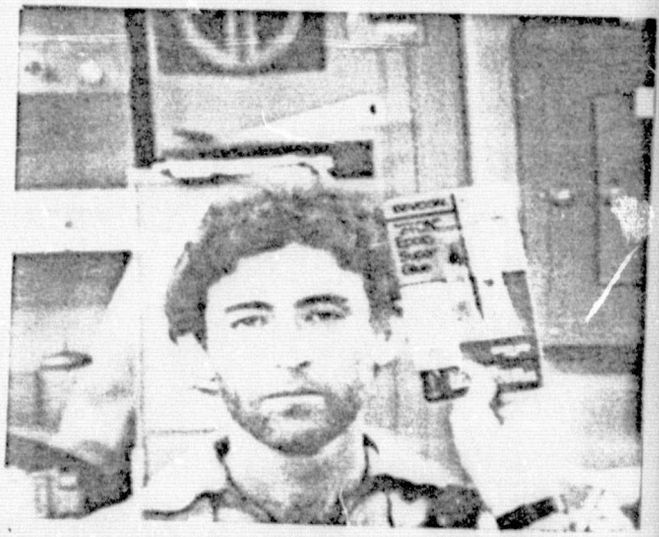
In terms of hardware the delta modulator with leaky integration is shown in block diagram form in Figure 29. Figure 29a shows the delta modulator without the leaky integrator and in Figure 29b leaky integration is achieved by putting a multiply by α block in the path of X_k. If $(1-\alpha)$ can be represented as $1/2^n$ as can 1/64 then the multiply by α can be implemented with a substractor. The subtractor subtracts X_k from itself shifted an appropriate number of times.

When leaky integrators are used, the DC shift in the decoder's estimate caused by a channel error will "leak" away exponentially with time. To understand how this happens refer to equation 5.2.1. The leak factor α decreases the term αX_k by the amount $(1-\alpha)X_k$, as compared with having no leak (α=1). To compensate for this decrease, the step size term, Y_{k+1}, will automatically on the average become slightly larger so that X_{k+1} remains unchanged. The increase in Y_{k+1} exactly

6 MB/S

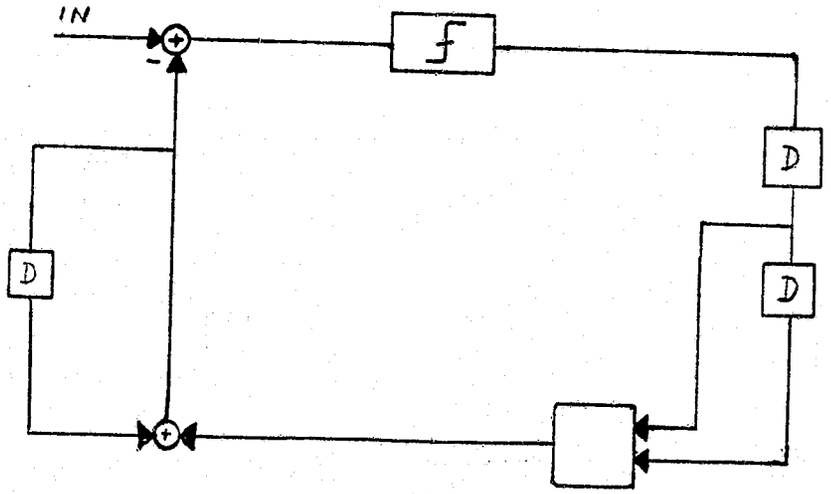
8 MB/S

16 MB/S

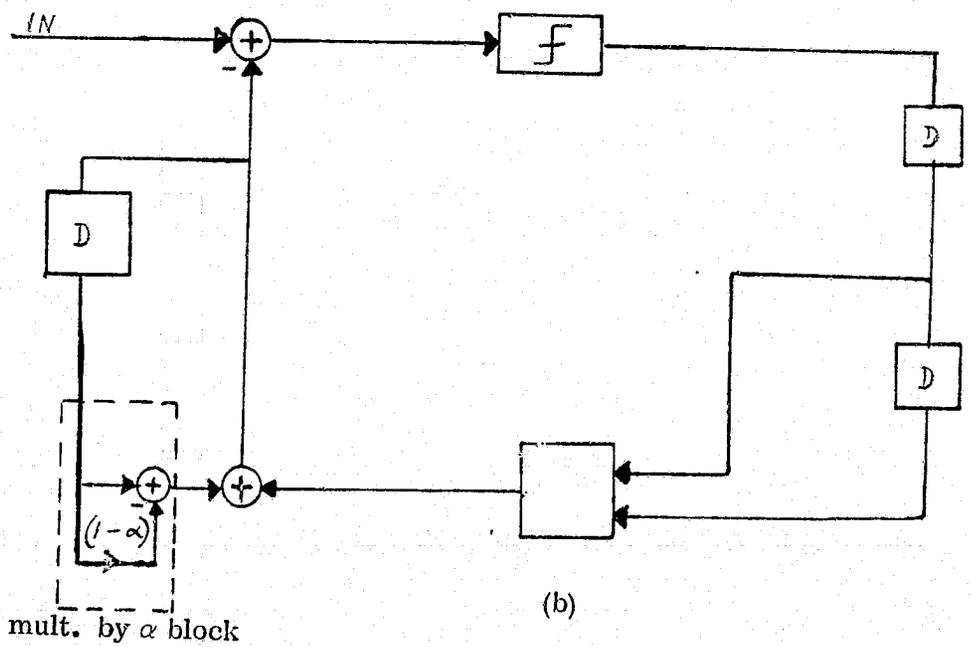


20MB/S

FIG 39 ONE DIMENSIONAL AMOD



(a)



mult. by α block

(b)

Fig. 27 (a) Delta modulator without leaky integrator
(b) Leaky integration achieved by including a multiply by α block in the path of Y_k.

cancels the "leak loss", $(1-\alpha)X_k$. Notice that the leak loss, $(1-\alpha)X_k$, depends on the amplitude of the estimate, X_k. If the decoder's estimate became larger than the encoder's estimate the leak loss at the decoder would be larger than the leak loss at the encoder and since the step size at the encoder and decoder are the same the decoder's estimate would decrease in amplitude until it equalled the encoder. Likewise, if the decoder's estimate became smaller than the encoder's estimate the leak loss at the decoder would be less than the leak loss at the encoder and thus the decoder's estimate would increase until it equalled the encoder.

Equation 5.2.1 can be solved recursively to determine the magnitude of the DC shift caused by a channel error as a function of time. If a channel error causes the estimate, X_k, to become shifted by an amount e at time k then at time k+1, k+2, . . . k+j, equation 5.2.1 becomes:

$$
\begin{aligned}
X_{k+1} &= \alpha(X_k + e) + Y_{k+1} \\
X_{k+2} &= \alpha(\alpha(X_k + e) + Y_{k+1}) + Y_{k+2} \\
X_{k+3} &= \alpha(\alpha(\alpha(X_k + e) + Y_{k+1}) + Y_{k+2}) + Y_{k+3} \\
&\vdots \\
X_{k+j} &= \alpha^j X_k + \alpha^j e + \sum_{i=1}^{j} \alpha^{j-i} Y_{k+i}
\end{aligned}
\qquad (5.2.2)
$$

From equation 5.2.2 we see that the error, e, will leak away by the factor α, and after j sample times the amplitude of the error will be $\alpha^j e$.

Clearly, the smaller the value of α the sooner the error will disappear. Small values of α will result in noticeably increasing the size of Y_k which will result in increasing the granular noise of the picture. From our computer simulations of leaky integrators shown in Figure 30 we have concluded that $\alpha = .97$ is the smallest leak factor that can be used without introducing annoying granularity into the picture.

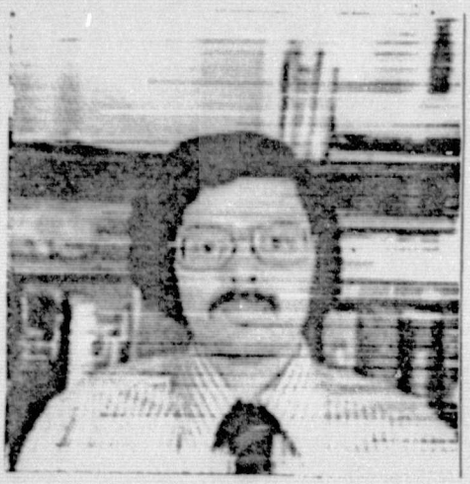
We will now repeat the analysis of section 4.2 to determine the mean square value of the DC shift caused by channel errors for a delta modulator which uses a leaky integrator for error correction.

The mean square value of $\overline{e^2}$ without the leaky integrator was found in equations 4.2.6 to 4.2.8a. To find $\overline{e^2}$ with the leaky integrator we must replace

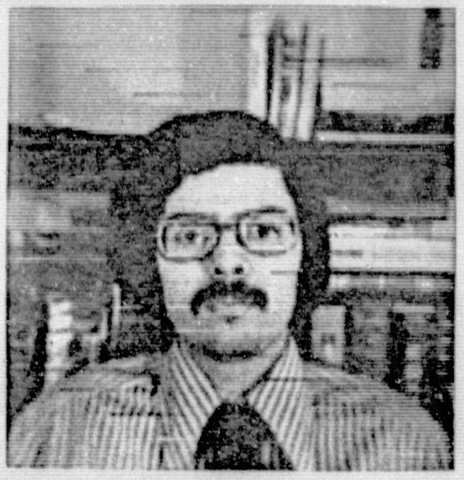
$$e_n(Y_k) \quad \text{by} \quad \alpha^j e_n(Y_k)$$

in equation 4.2.6. Carrying this change through to equation 4.2.8a and calling the mean square error with leaky integration, $\overline{e_\ell^2}$, we find that

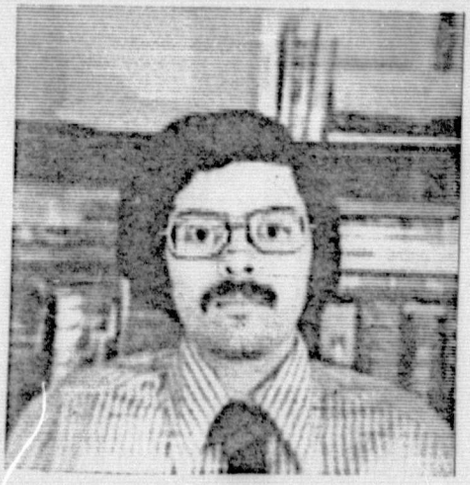
$$\overline{e_\ell^2} = \alpha^{2j}\overline{e^2} \tag{5.2.3}$$



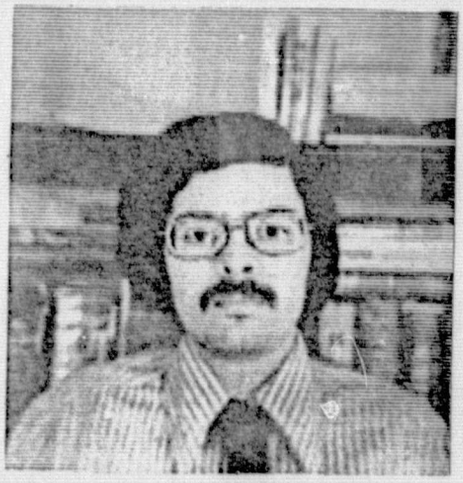
No Leak



Leak Factor .99



Leak Factor .98



Leak Factor .875

Fig 30 The effects of various leak factors on the delta modulated pictures in the presence of channel errors.

where $\overline{e^2}$ is the error without the leaky integrator. Thus the mean square error with a leaky integrator can be calculated from equation 4.2.8a and 5.2.3 given the step size statistics and E_k pattern statistics for a delta modulator with leaky integration.

The step size and E_k pattern statistics for a delta modulator with leaky integration is given in Table IV. From equation 4.2.8a, 5.2.3 and Table IV the mean square error is found to be

$$\overline{e_\ell^2} = \alpha^{2j}(179) \tag{5.2.4}$$

where "j" is the number of sample times after the occurrence of the channel error and α is the leak factor.

Proceeding as in section 4.2 the energy per error D_e is given by

$$D_e = \frac{1}{N}\sum_{i=0}^{\infty} \alpha^{2i}(179) \tag{5.2.5}$$

where N is the number of samples per pel. We multiply by 1/N so that in converting power to energy the unit of time in equation 5.2.5 remains the pel. This makes equation 5.2.5 consistent with the equations of section 4.2 and 5.1.

Np, the distortion energy per pel, is given by

$$Np = D_e PeN$$

TABLE IV

$\lvert Y_k \rvert$	$P(\lvert Y_k \rvert)$	n - E_k's	P(n)
1	.24	1 - 1100	.17
2	.32	2 - 1001	.14
3	.23	3 - 1101	.11
4	.12	4 - 1000	.15
5	.0016	5 - 1010	.04
6	.05	6 - 1110	.15
7	.0038	7 - 1011	.11
8	0	8 - 1111	.12
9	.017		
10	.0016		
11	0		
12	0		
13	.0065		
14	0		
15	.0049		

Step size and E_k pattern statistics for a delta modulator with leaky integration

Substituting from equation 5.2.5 yields

$$Np = \frac{Pe(179)}{1-\alpha^2} \tag{5.2.6}$$

From equation 4.2.12 and 5.2.6 and the fact that Sp/Np is set equal to 30 db we have

$$\frac{Sp}{Np} = \frac{q^2/3}{Pe(179)/(1-\alpha^2)} = 1000 \tag{5.2.7}$$

Using the parameter values of section 4.2 and α = .97, Pe is found from equation 5.2.7 to be

$$Pe = 4.6 \times 10^{-4} \tag{5.2.8}$$

Pe is the probability of error required to obtain a -30 db distortion level due to channel errors. In comparing Pe from equation 5.2.8 with the Pe from equation 4.2.14 we see that we can tolerate 38 times more channel errors when leaky integration is used as compared to the delta modulator without leaky integration.

From the probability of error we can find the transmitter power as in section 4.2, which gives -30 db distortion with leaky integration. Thus substituting into the following equations from section 4.2

$$Pe = 4.6 \times 10^{-4} = Q(\sqrt{2E_b/n_o})$$

$$E_b/n_o = 7.5 \text{ db}$$

$$S_T = E_b N(2fm)$$

yields

$$\boxed{\frac{S_T}{n_o fm} = 15.3 \text{ db}} \qquad (5.2.9)$$

Comparing equation 5.2.9 with 4.2.19 shows that the leaky integrator can save 2 db of transmitter power compared with the delta modulator without leaky integration.

5.3 Line to Line Correlation

Another kind of error correction is one which employs an algorithm whereby the receiver can examine the received bit sequence and determine the transmitted sequence inspite of channel errors in the received sequence. Such an algorithm can be found provided that the transmitted bits are correlated to each other in some manner.

There is a great deal of line to line correlation in pictures; that is to say that any scanning line tends to be similar to the line above and below it over much of the line. Exceptions to this will occur when

a scanning line coincides with a perfectly horizontal edge of a large object. In such a case the line would be correlated only to the line above it or below it but not both. Another exception would be a long perfectly horizontal strip just one scanning line thick. In this case the scanning line on the horizontal strip would be different from both the line above and below it. Both of these exceptions are relatively rare in a picture, the latter case is so unlikely that it would not even occur once in a typical picture.

When a channel error occurs the line containing the error becomes uncorrelated with the line above and below it. This lack of correlation appears as a streak as seen in Figure 21b. Thus when no channel errors occur, the delta modulator's estimates, X_k, (and therefore E_k) are highly correlated in the vertical direction, and when a channel error occurs they become uncorrelated.

This situation is analogous to channel coding for error correction. In channel coding uncorrelated information bits are correlated by adding extra bits, in for example, a cyclic encoder. When a channel error occurs the correlation is lost, and a sequence of bits is received that the transmitter could never have sent. A tree type decoder stores all possible transmitter sequences and assigns a weight to each sequence according to how unlikely it was for the transmitter to have transmitted that sequence. Usually, the likelihood of a transmitter sequence is based upon how many bit errors must be assumed in the received sequence for the received sequence to match the transmitter sequence. The

decoder, at the end of the message, chooses the transmitter sequence which required the smallest number of channel errors as the sequence actually transmitted, i.e., the sequence with the smallest weight is chosen.

For the delta modulator, as in channel coding, a weight can be assigned to each possible transmitter sequence of E_k's that is inversely related to the probability of that sequence as calculated from the received sequence. The weight assigned to each transmitter sequence has two parts. The first part depends upon how many channel errors we must assume occurred in the received sequence to make it match a transmitter sequence. Transmitter sequences that differ from the received sequence in a few bits will have lower weight than transmitter sequences that differ in many bits. Let us call this weight, W_b, the weight due to bit differences between received and transmitter sequences.

The second part of the weighting function (call it Wc) derives its value from the amount of uncorrelation between the adjacent scanning lines and each possible transmitter sequence. Thus a transmitter sequence of E_k's that results in generating a line of estimates, X_k, that are similar to the line of estimates above or below it will have a small weight. Those transmitter sequences that result in generating a line of estimates that are dissimilar to the line of estimates above or below it will be assigned large weights.

The total weight assigned to a transmitter sequence is

$$W_T = W_b + W_c \tag{5.3.1}$$

If, as the received sequence is compared bit by bit with a transmitter sequence we assign a weight 1/Pe each time they disagree in a bit and a weight 1/(1-Pe) each time they agree then W_b is given by:

$$W_b = \frac{n}{Pe} + \frac{\ell - n}{1 - Pe} \tag{5.3.2}$$

ℓ ≡ Number of bits in a sequence which is also the number of E_k bits per line
n ≡ Number of bits in which the received sequence differs from a transmitted sequence
Pe ≡ Probability of a channel error

Notice that a Pe = .5 results in a value of W_b which is equal to 2ℓ and independent of n and hence, the received sequence. At the other extreme Pe = 0 results in an infinite weight being assigned to any transmitter sequence that differs from the received sequence in any bit. W_b ranges in value from ℓ to infinity.

The weight, W_c, is found for each transmitter sequence by subtracting each estimate of each transmitter sequence from the corresponding estimate in the scanning line above and below it. The resulting difference signal, d, is given by:

$$d = \text{min of} \begin{matrix} |X_k - X_{k-\ell}| \\ \text{or} \\ |X_k - X_{k+\ell}| \end{matrix} \tag{5.3.3}$$

It is assumed that because of line to line correlation the estimate X_k will be nearly equal to $X_{k-\ell}$ or $X_{k+\ell}$ and d will be a random variable with a small variance. When a channel error occurs, however, d will become large and assume a value of low probability. Let $p_d(d)$ be the density function of d, then a good choice for W_c is

$$W_c = \frac{2}{q} \sum_{i=0}^{\ell} \frac{1}{p_d(d_i)} \tag{5.3.4}$$

where q is the number of quantization levels in the estimate.

Notice that if d had a uniform distribution the value of W_c would be independent of d and equal to 2ℓ, a situation analogous to equation 5.3.2 for Pe = .5. If perfect correlation existed between lines in a picture then $p_d(0) = 1$, $p_d(d) = 0$, $d \neq 0$ and any transmitter sequence which had a $d \neq 0$ would have an infinite weight assigned to it. This situation is analogous to Pe = 0 in equation 5.3.2.

If an error occurs near the end of a line the weight W_c may be too small to properly weight the correct sequence because only the last few terms in the summation of equation 5.3.4 would have large values for the received (but incorrect) sequence. To counter this problem a weighting function g(i) might be introduced to give extra emphasis to d_i's near the end of the line. Thus we might make g(i) equal to

$$g(i) = ba^i \tag{5.3.5}$$

where a and b are constants (close to unity) whose value would determine the amount of weight g(i) had on W_c. Substituting equation 5.3.5 into 5.3.4 yields

$$W_c = \frac{2b}{q}\sum_{i=0}^{\ell}\frac{ba^i}{p_d(d_i)} \tag{5.3.6}$$

Combining equations 5.3.1, 5.3.2 and 5.3.6 yields:

$$W_T = \frac{n}{Pe} + \frac{\ell-n}{1-Pe} + \frac{2b}{q}\sum_{i=0}^{\ell}\frac{a^i}{p_d(d_i)} \tag{5.3.7}$$

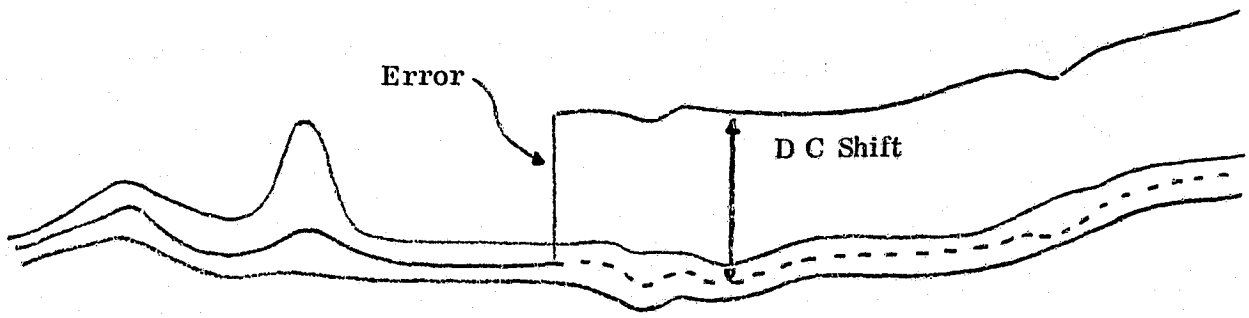
Using equation 5.3.7 we can assign a weight to each transmitter sequence. Unfortunately there are 2^{1500} possible transmitter sequences since there are 1500 bits per line. If we use Viterbi's tree searching algorithm then the number of possible sequences that the Viterbi decoder will have to search is reduced to the number of states that the delta modulator can assume. The delta modulator has 6 bits in the estimate, 4 bits in the step size and one bit for E_{k-1}; therefore, the number of sequences to be searched is 2^{11} or 2048 sequences. It is possible to search 2048 sequences but it is not practical to do it in real time. Since our objective is to design a practical real time video encoder we abandoned this approach for a much simpler approach to error correction which uses the correlation between lines but does not use a tree

searching algorithm.

The technique that we employed detects the occurrance of an error by assuming that horizontal streaks are errors. The visibility of the error is reduced by replacing the DC level of the streak by the DC level of the average of the line above and below it.

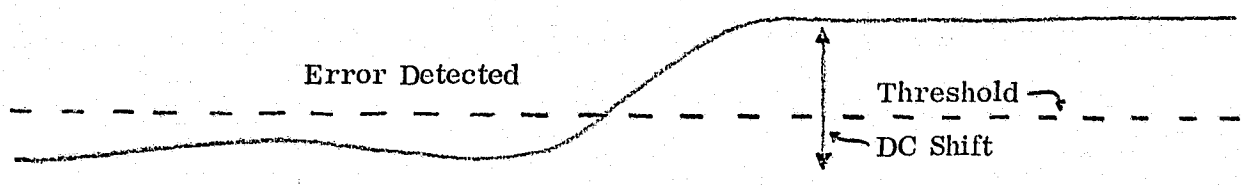
The technique is shown in Figure 31. Figure 31a shows three consecutive scanning lines. The middle scanning line has an error and, therefore, its DC level has shifted. N consecutive pels for each line are then averaged as shown in Figure 31b so that the DC level for the N pels can be determined. The amplitude of the middle line in Figure 31b is compared with the line above and below it. If the amplitude of the middle line differs from the line above it and below it by more than a certain threshold an error streak is assumed to be present. From the point at which an error is detected to the end of the line, the DC value of the line with the streak is replaced with the average DC value of the line above and below it as determined by Figure 31b. The result of doing this is shown in Figure 31d.

There are two parameters in the above algorithm that must be adjusted, "N" and the "threshold", to eliminate the error streaks. In Figure 32 we can see the effects of "N" and the "threshold" on both the error streaks and the quality of the picture. Small values of "N" and low thresholds eliminate the error streaks but they tend to degrade the picture by eliminating thin objects such as the bottom of my glasses as shown in Figure 32e. Large values of "N" and high thresholds leave

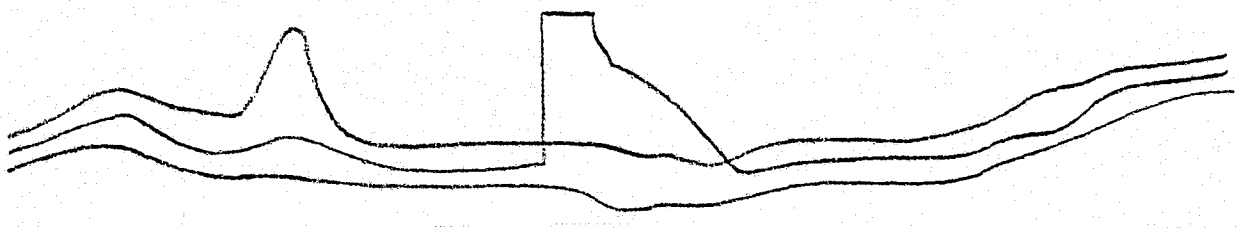


Part A. The Video Signal From 3 Consecutive Scanning Lines

Part B. The Video Signal After Averaging 'N' Consecutive Pixels

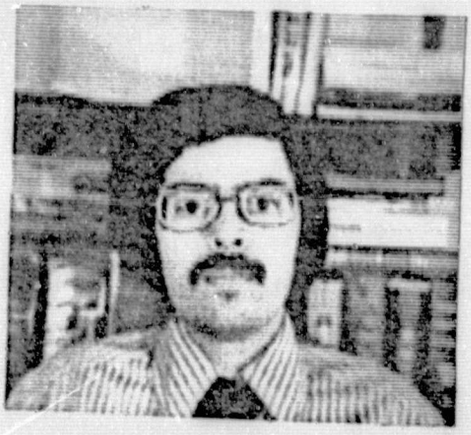


Part C. The Absolute Value of the Difference Between the Middle Wave Form and the Average of the Top and Bottom of Part B. is shown in Part C.



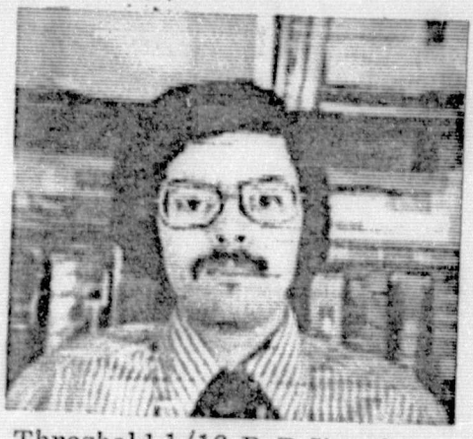
Part D. The D C Shift of Part C. is Subtracted from the Video Signal to Produce the the Corrected Signal of Part D.

Fig. 31 A Line to Line Correlation Algorithm for Error Correction

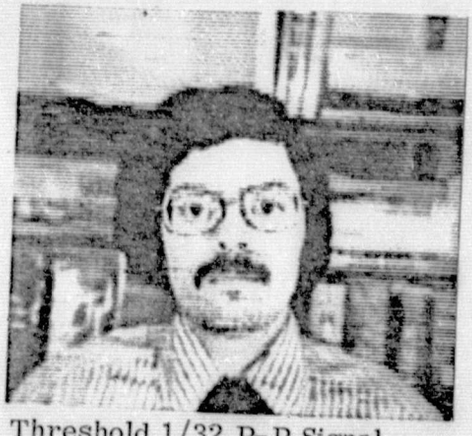


Original Picture
(a)

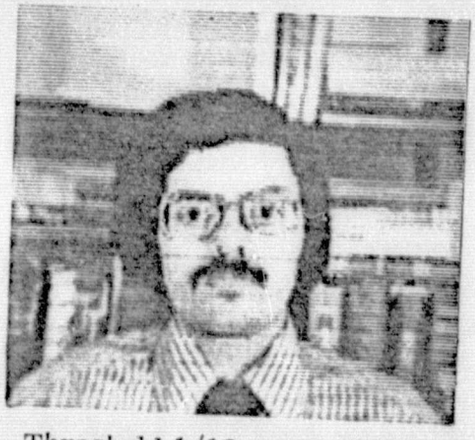
Noisy Picture
(b)



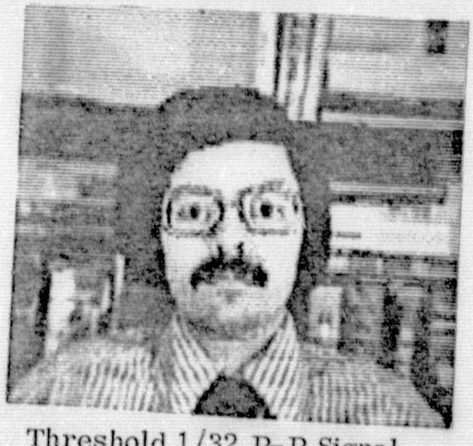
Threshold 1/16 P-P Signal
Averaged Over 1/16 Of A Line
(c)



Threshold 1/32 P-P Signal
Averaged Over 1/16 Of A Line
(d)



Threshold 1/16 P-P Signal
Averaged Over 1/128 Of A Line
(e)



Threshold 1/32 P-P Signal
Averaged Over 1/128 Of A Line
(f)

Fig 32 The results of using the line to line correlation algorithm on noisy pictures. 1000 bits/line; 170 lines per picture; error rate of 3×10^{-3}

some error streaks undetected and parts of others uncorrected but they do not degrade the picture. It is our opinion that Figure 32c represents the best compromise between the degree of error correction necessary and the amount of degradation that can be tolerated in the picture.

5.4 A Real Time Delta Modulator with Error Correction

Of the three error correction algorithms investigated the leaky integrator error correction technique gave the best results and is also the easiest to implement.

We designed a real time delta modulator using a leaky integrator for error correction. A block diagram is shown in Figure 33. The leak factor α was chosen to be .968. If the estimate before the leak is called X_k and if the estimate after the leak is $X_k^{(\ell)}$ then the equation defining the leaky integrator is given by:

$$X_k^{(\ell)} = .968X_k + 2$$

$$X_k^{(\ell)} = X_k - \frac{X_k}{32} + 2 \qquad (5.4.1)$$

Since X_k assumes values between zero and 127, the addition of the value of 2 to equation 5.4.1 causes X_k to leak symmetrically about the midvalue 63. As an example consider that when X_k equals zero, $X_k^{(\ell)}$

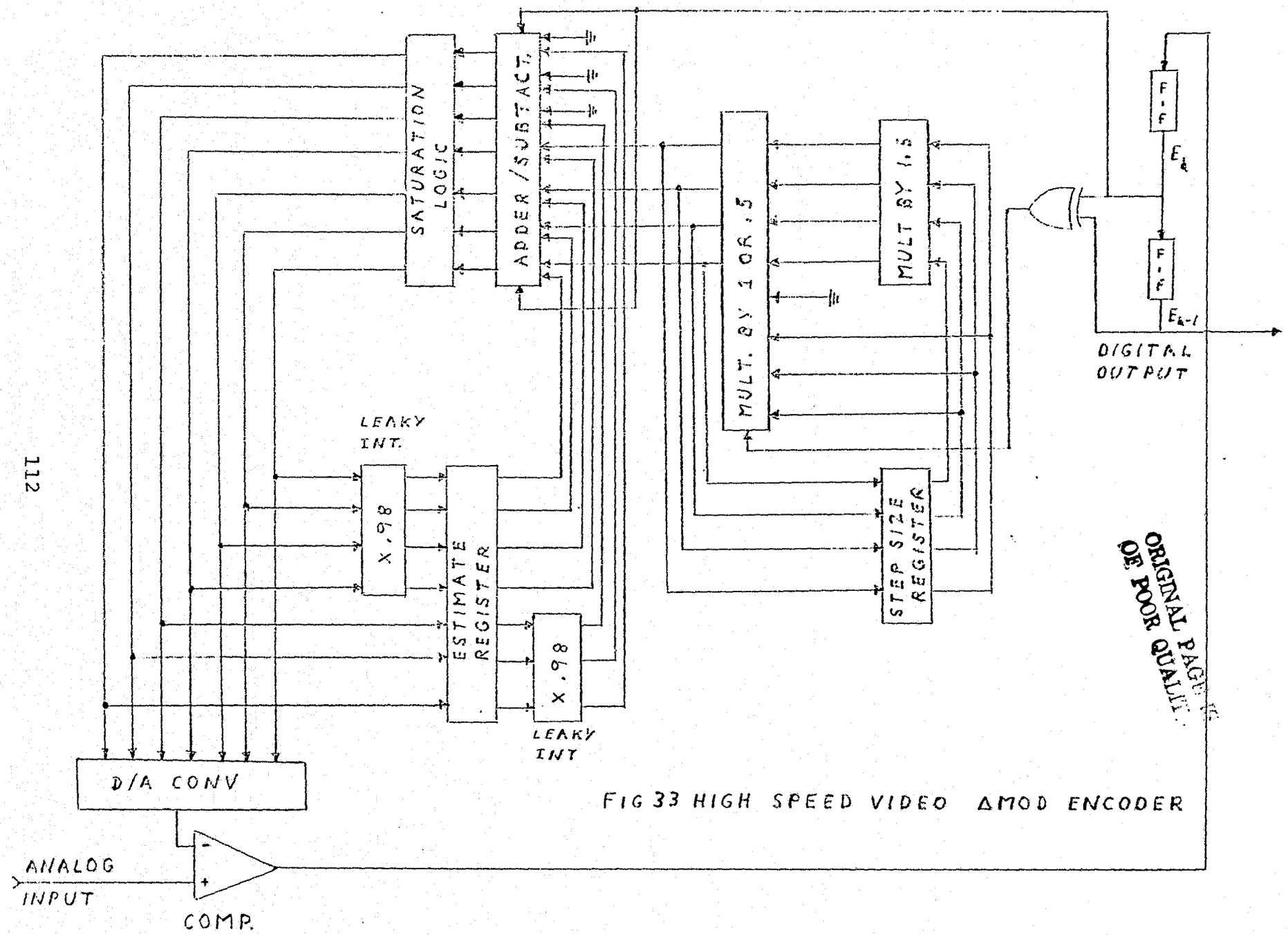


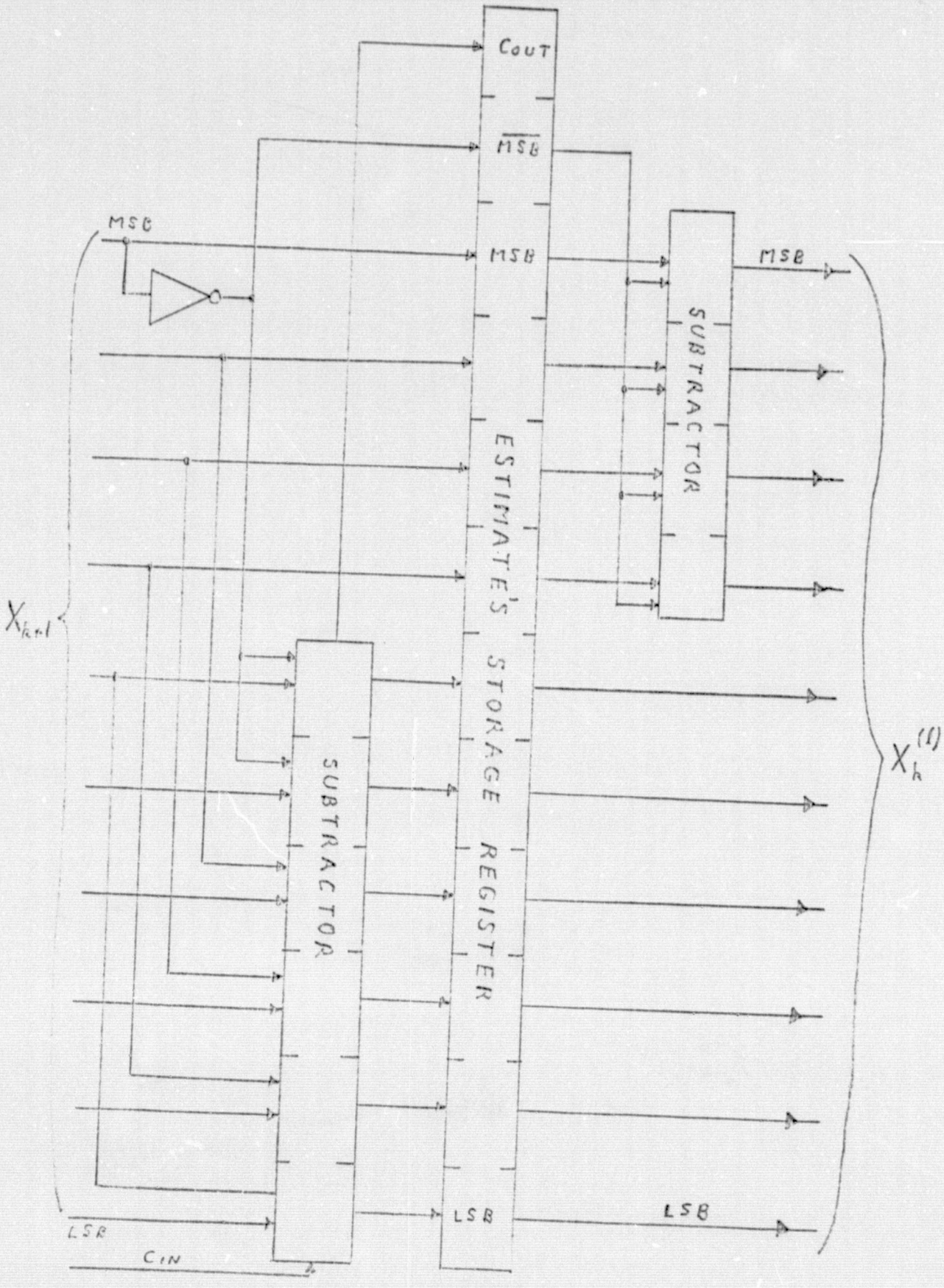
FIG 33 HIGH SPEED VIDEO ΔMOD ENCODER

will leak up to the value +2, when X_k equals 127 $X_k^{(\ell)}$ will leak down 2, to the value +125, and when X_k assumes the midvalue 63 $X_k^{(\ell)}$ will remain unchanged. The addition of the number 2 minimizes the distortion caused by leaking X_k while having no effect on the leaking away of errors.

The implementation of the leaky integrator is shown in Figure 34. The subtraction of $X_k/32$ from X_k is carried out in Figure 34 by shifting X_k five places and subtracting it from X_k. The addition of +2 in equation 5.4.1 is carried out by the same subtractor by using the inverted value of the most significant bit of X_k as the six most significant bits of $X_k/32$. Also notice that the 6 least significant bits leak before the storage register and the 4 most significant bits leak after the storage register. In order to split up the leaky integrator this way it is necessary to store the carry out (C_{out}) from the subtractor that leaks the 6 least significant bits so that it will appear at the carry in (of the subtractor that leaks the 4 most significant bits) at the same time that the estimate appears at the subtractor's inputs.

The leaky integrator was split up this way (half the leak on X_{k+1} and half on X_k) to minimize the effect of the carry propagation delay through the substractor on the sampling rate of the delta modulator. Before any storage register's can be clocked the C_{out} signal must be present and (see Figures 34) the comparator's output must have settled down. With only 6 bits to propagate through, the C_{out} signal will appear before the comparator's output settles down; therefore, the subtractor

Fig. 34 Leakey Integrator



to the left of the storage register is not part of the maximum delay time path of the delta modulator and will not affect the sampling rate of the delta modulator. If the subtractor to the right of the storage register (in Figure 34) is to be part of the maximum delay time path, the propagation delay of the subtractor would have to be greater than that of the step size generator. This is not the case; hence, the leaky integrator split up as in Figure 34 will not affect the sampling rate of the delta modulator. If the leaky integrator was placed entirely to the left or right of the storage register then the carry propagation delay would be sufficient to reduce the maximum sampling rate of the delta modulator.

Because the leaky integrator has only 10 bits and not 12 the effects of a channel error may not leak away completely. Under worst case conditions the estimate at the delta modulator decoder could remain 3 quantization levels greater than, or less than, the estimate at the encoder. To remove the last vestige of channel errors the estimate at the encoder and decoder are both forced to their minimum value of zero at the end of each scanning line. Also, the step size is forced to its maximum value of 15 and E_{k-1} is forced to -1 in both the encoder and decoder at the end of each scanning line. This ensures that the encoder and decoder begin each scanning line in the same state, free of any remaining channel errors.

The conditions described above are forced upon the delta modulator in a very simple fashion. The negative going sync pulse on the composite

video signal from the TV camera is set to a voltage more negative than the most negative output of the D/A converter of Figure 33. When the sync pulse occurs the estimate is driven to the zero quantization level and held there by the saturation logic. Since the sync pulse is more negative than the zero quantization level the comparator of Figure 33 outputs a minus one E_k during the sync time. The string of minus one E_k's are fed back to the step size generator and causes the step size to grow to its maximum value.

The complete circuit diagrams for the delta modulator encoder and decoder are shown in Figures 35, 36, 37 and 38. Figure 35 shows the input signal from the TV camera being band limited to 4 MHz, DC restored, buffered, and then clamped to 1 volt peak to peak. The output of Figure 35 is fed into the input of Figure 36. Figure 36 is the complete circuit schematic of the delta modulator encoder with error correction. Figure 37 is the circuit schematic of the delta modulator decoder. Figure 38 is the detailed schematic of the analog output of the decoder. It shows the circuitry which restores the sync pulses to the composite video signal. The delta modulator was built out of ECL 10,000 series logic and it has a maximum clock rate of 23 MHz.

A series of tests were made with the delta modulator. The results of the first test are shown in Figure 39. Here we see the effects of the delta modulation of the 4 MHz composite video signal at clock rates of 6, 8, 16 and 23 MHz. The pictures are fairly good at 16 MHz and almost of broadcast quality at 23 MHz. For the second test we

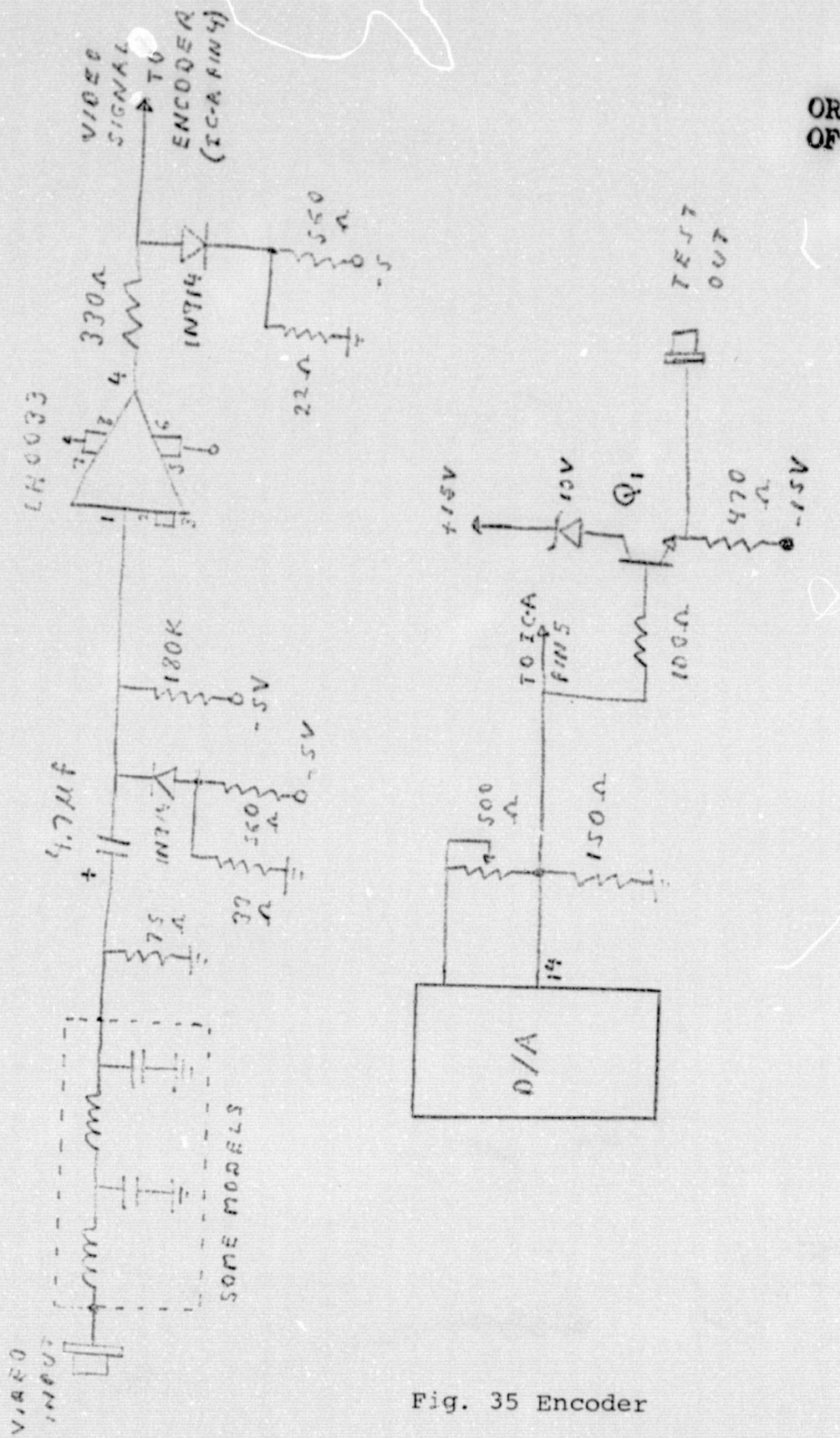


Fig. 35 Encoder

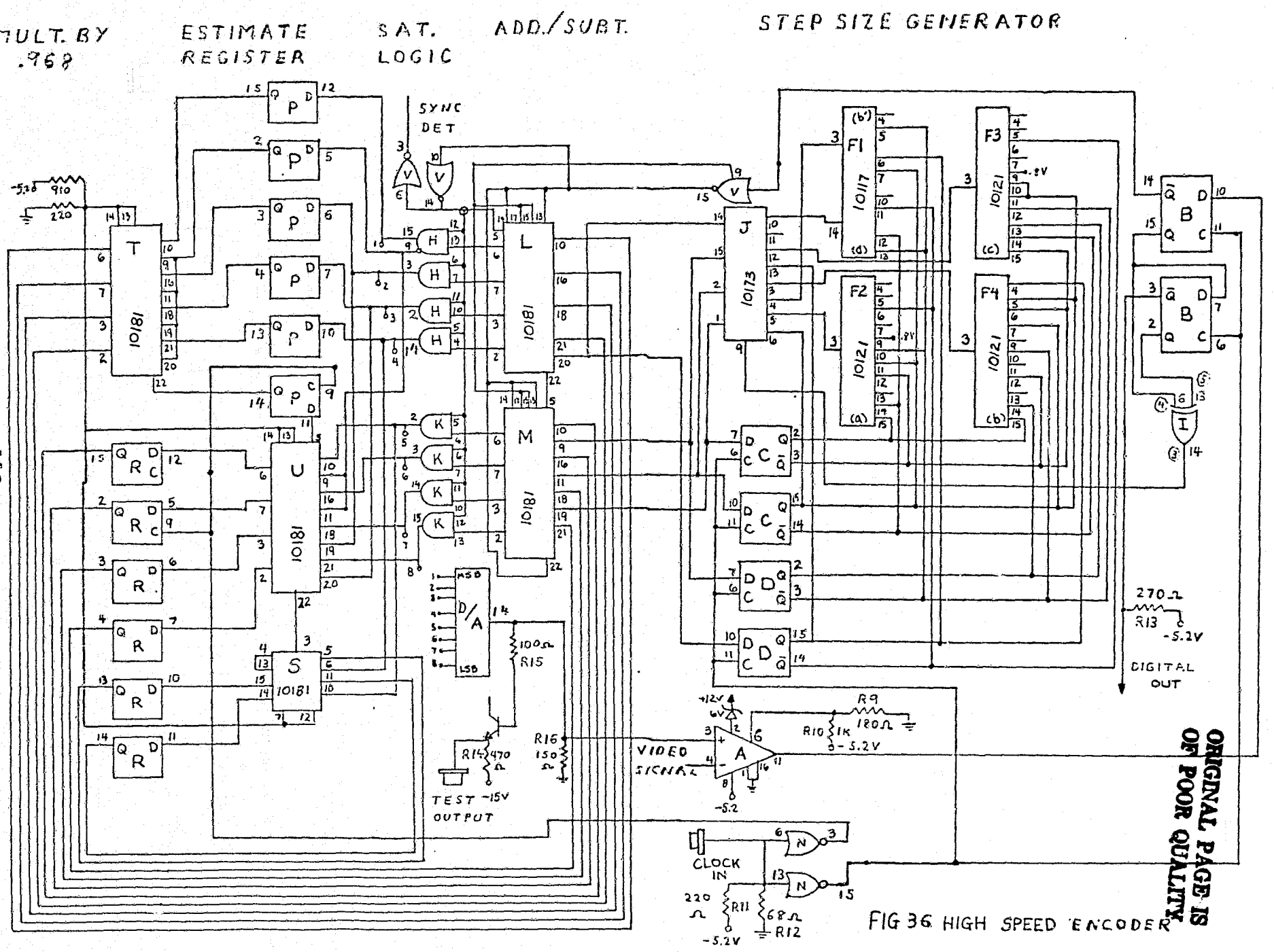


FIG 36 HIGH SPEED ENCODER

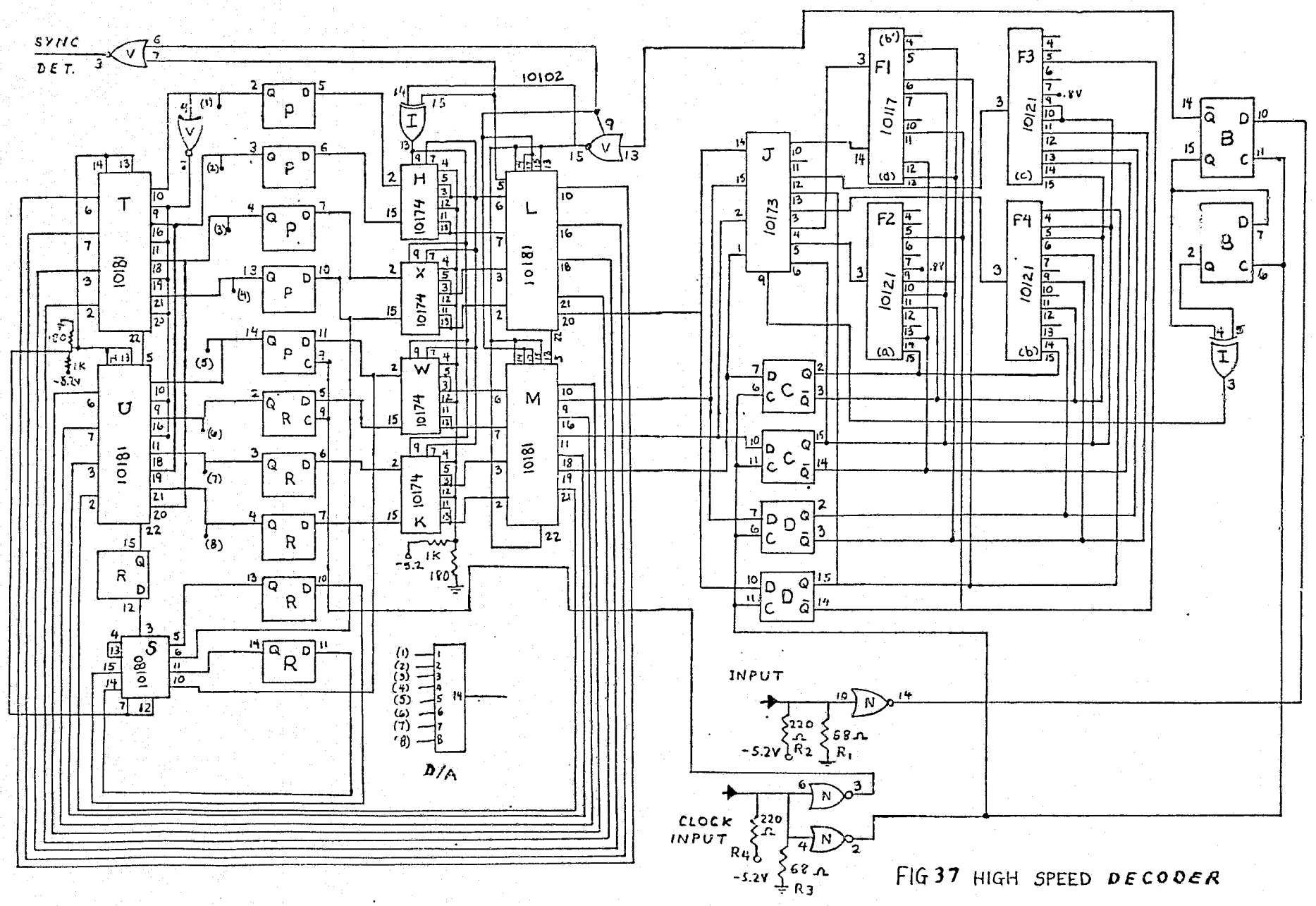


FIG 37 HIGH SPEED DECODER

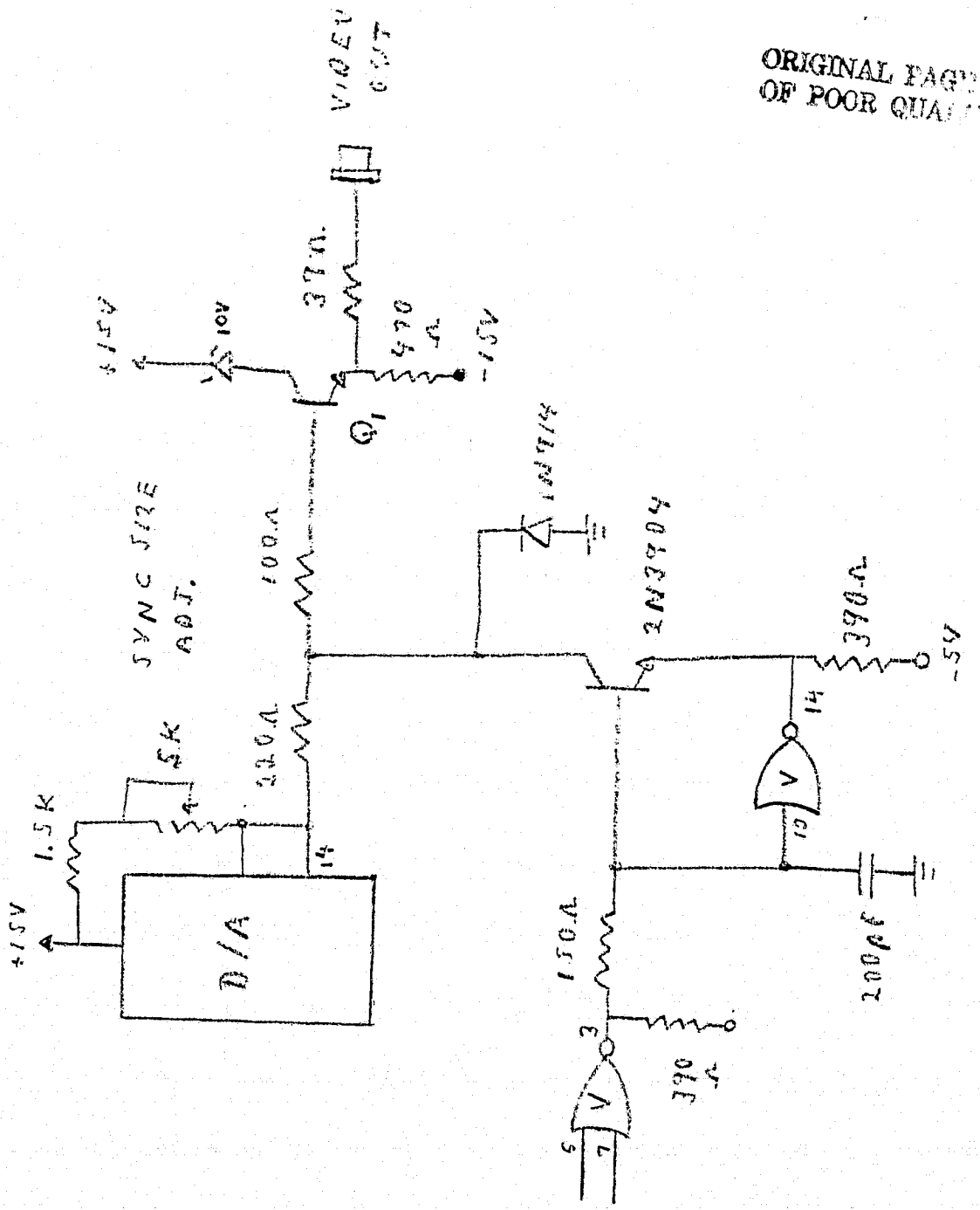
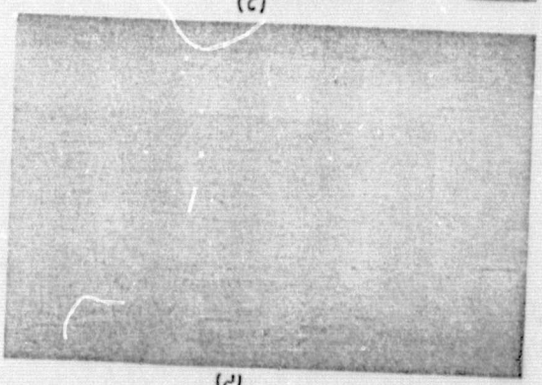


Fig. 38 Decoder

(a)

(b)

(c)



(d)

Fig. 40 Response of ADM to channel errors.

(a) No errors (b) $P_e = 10^{-3}$ (c) $P_e = 10^{-2}$
(d) $P_e = 10^{-2}$

introduced errors at 10^{-4}, 10^{-3} and 10^{-2} errors per bit. The pictures of Figure 40 show the effects of errors. At 10^{-4} errors per bit the errors are almost unnoticeable. At 10^{-3} the errors are noticeable but not annoying. At 10^{-2} errors per bit the picture information is stil intelligible but the errors are very annoying.

5.5 Conclusions

The delta modulator described in section 5.4 fulfills the objectives of this study. The delta modulator can digitize a video signal while providing nearly error free pictures with bit error rates as high as 10^{-4} errors per bit. The delta modulator is also cost effective. It consists of only 21 IC's and consumes only 12 watts of power. Its only shortcoming lies in the fact that its bandwidth compression is only 2:1 for undistorted encoding of TV pictures.

CHAPTER VI

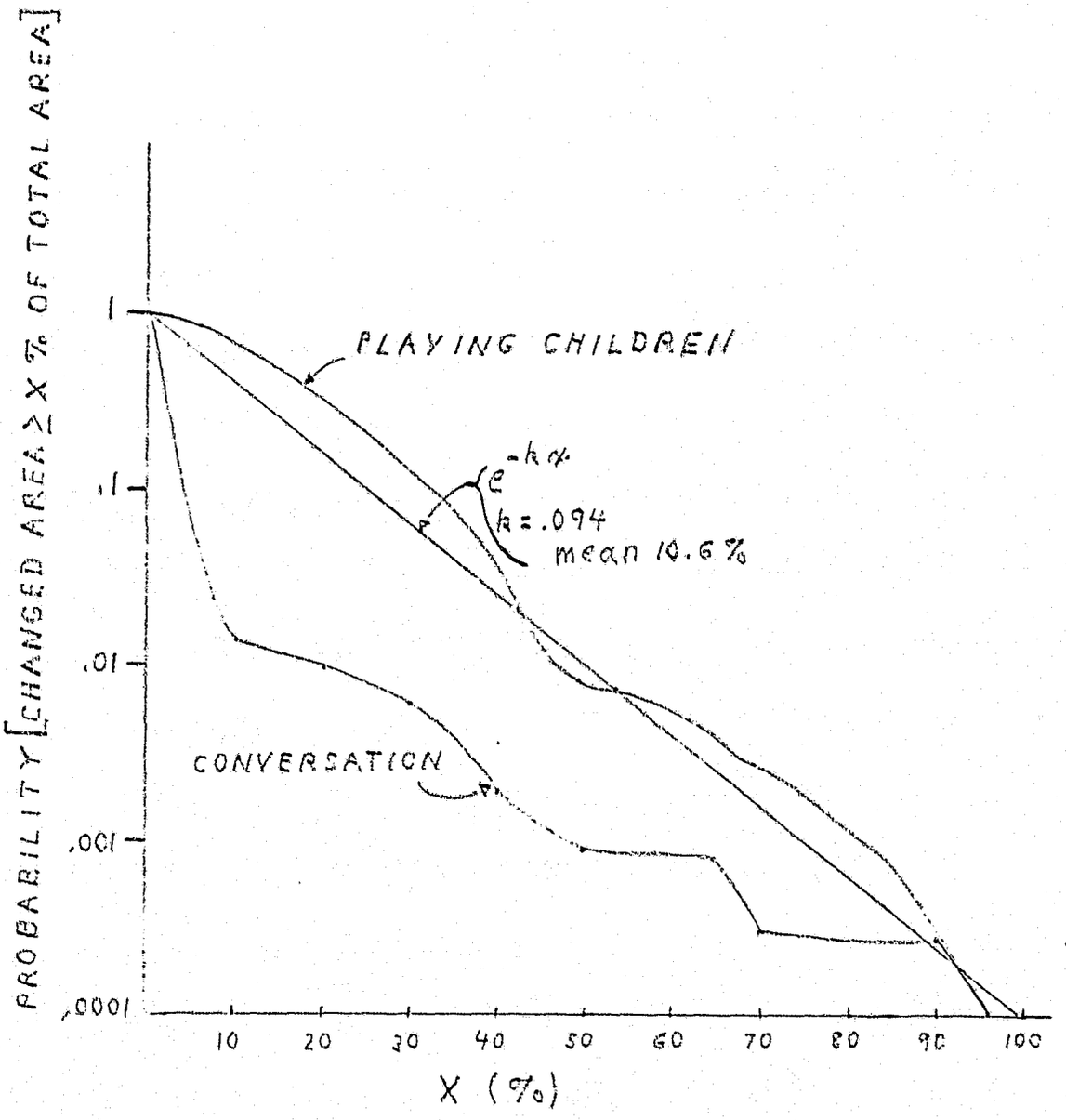
Other Delta Modulators

To increase the bandwidth compression ratio of the delta modulator while also decreasing the distortion caused by the delta modulator, we investigated several types of delta modulators substantially more complex than the one described in section 5.4. The first delta modulator investigated was an interframe encoder.

6.1 Interframe Delta Modulation

From the previous work presented, it is apparent that the key to producing high quality pictures at low bit rates, with delta modulation, lies in reducing edge business. In this section we explain an interframe encoding technique employing a delta modulator which will reduce edge business in real time motion pictures.

Successive frames of a motion picture contain redundant information. The amount of redundancy depends upon the degree of change in the scene from frame to frame. Figure 41 shows the statistics on the amount of movement in broadcast television (9) for playing children and conversation. An exponential distribution can be fit to the data of Figure 41 with a mean of 10.6%. This means that on the average only 10.6% of the pels change from frame to frame. It is possible to design a delta modulator which will produce good pictures at low bit rates by taking advantage



$X = \frac{\text{NUMBER OF CHANGED PELS}}{\text{TOTAL NUMBER OF PELS}}$

FIG 41 STATISTICS ON THE AMOUNT OF MOVEMENT IN BROADCAST TELEVISION (9)

of the interframe redundancy of motion pictures.

Figure 42 shows the encoding scheme that the interframe delta modulator will use. Each large square represents a successive frame of a motion picture. Each dot on a frame represents a pel (picture element). Note that each pel has associated with it a delta modulator (ΔMOD) that follows the same pel through sucessive frames. Thus the ΔMOD in the upper most left hand corner always encodes the pel in the upper most left hand corner for every frame.

The encoding of high quality pictures at a low bit rate is achieved in the following manner. From previous studies of delta modulators it has been observed that high sampling rates are required for a delta modulator to accurately encode rapidly changing signals and low sampling rates may be used on slowly varying signals. The usual method of encoding a picture by sampling successive pels within the same frame results in rapidly changing signals which require a high delta modulator bit rate. With the frame to frame encoding technique of Figure 42 each delta modulator is associated with its own pel. On the average, 90% of the pels will not change value from frame to frame, allowing the delta modulator to accurately encode the value of the pel at a low bit rate. Of course, those pels that do change value will not be encoded accurately. Using this scheme motion pictures will be encoded at 1 bit per pel.

At first thought, hardward implementation of Figure 42 may seem impossible since, at one delta modulator per pel, a typical 250,000 pel/frame 4 MHz bandwidth TV system would have to contain 250,000 delta

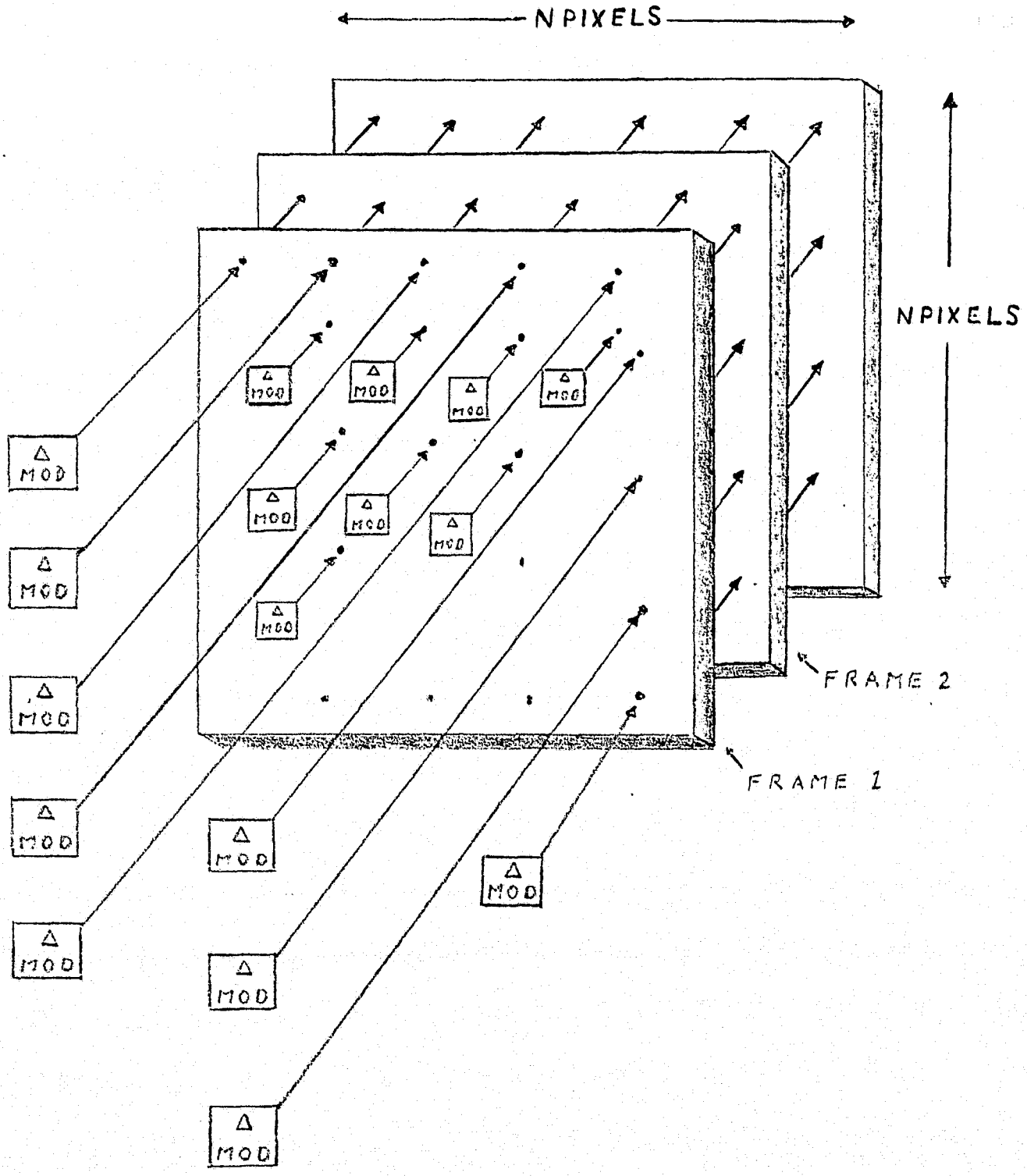


FIG. 2 FRAME TO FRAME ENCODING OF PICTURES USING DELTA MODULATION

modulators each operating at a 30 Hz sampling rate. Fortunately only one 8 MHz delta modulator need be used to implement Figure 42; however, this single delta modulator will have to contain enough shift register type memory to store an entire picture frame.

The reason that one delta modulator can replace all the delta modulators of Figure 42 is simple. If Figure 42 was implemented as shown all 250,000 pels could be encoded, decoded and displayed in parallel, but a real time television system requires only one pel at a time in serial. The retention time of the eye and screen give the appearance of a full picture. We may take advantage of this by starting a single delta modulator at the upper most left hand corner of a frame, let it encode that pel based upon its previous estimate of the pel from the previous frame, transmit a bit (E_k), store its new estimate (X_k) for that pel, and then repeat the process for the next adjacent pel in the same frame. The delta modulator will continue to encode, and transmit for each adjacent pel in turn, until the delta modulator has been multiplexed through the entire frame (typically 1/30 sec). Then the delta modulator will return to the first pel and repeat the process for the next frame.

Figure 43 shows the block diagram of the interframe delta modulator. This delta modulator is the same as the one shown in Figure 16 except for the addition of twelve 250,000 bit long shift registers. The shift registers were assembled out of 4K dynamic random access memories.

The following paragragh gives a brief explanation of how the memories

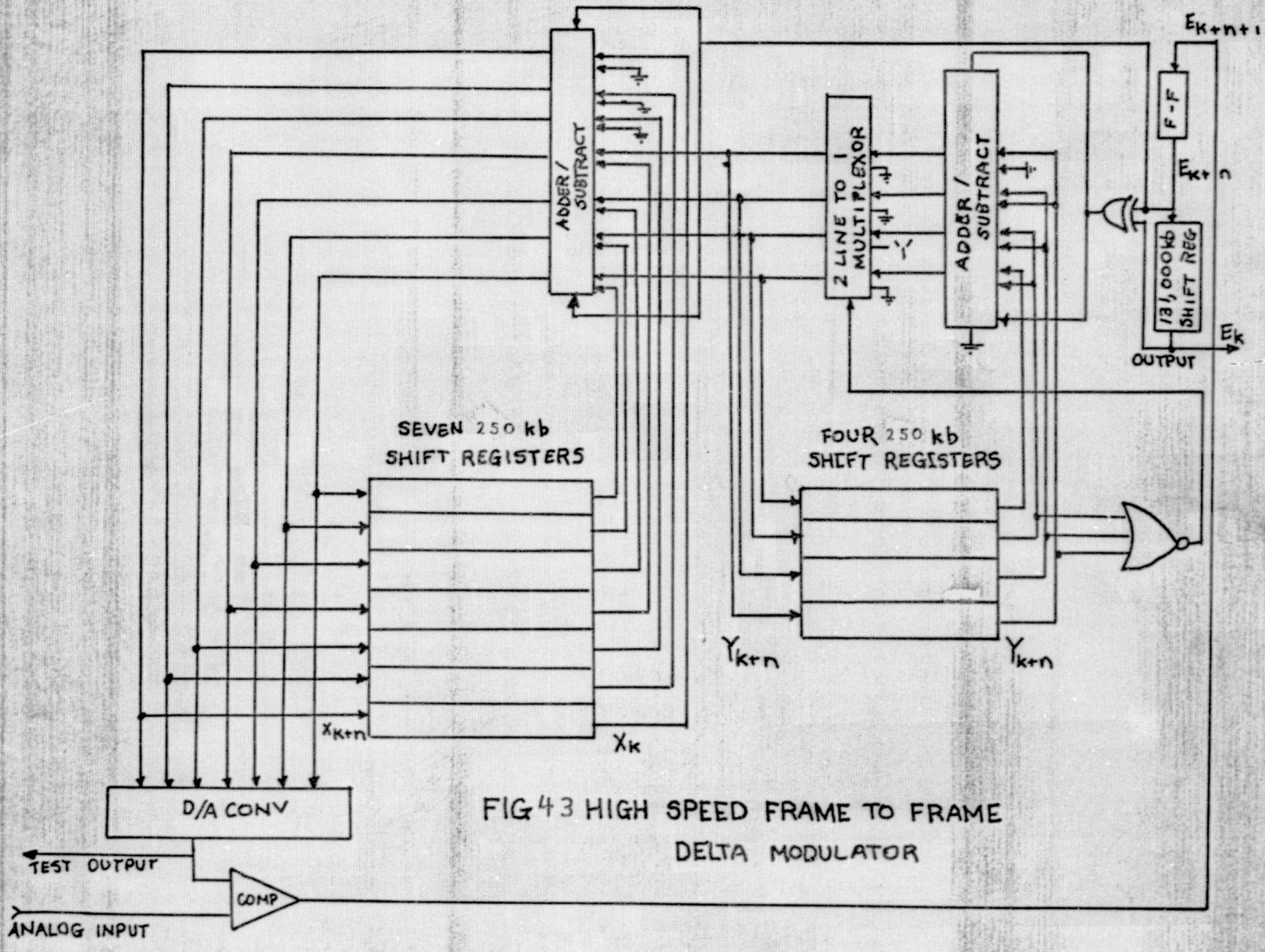


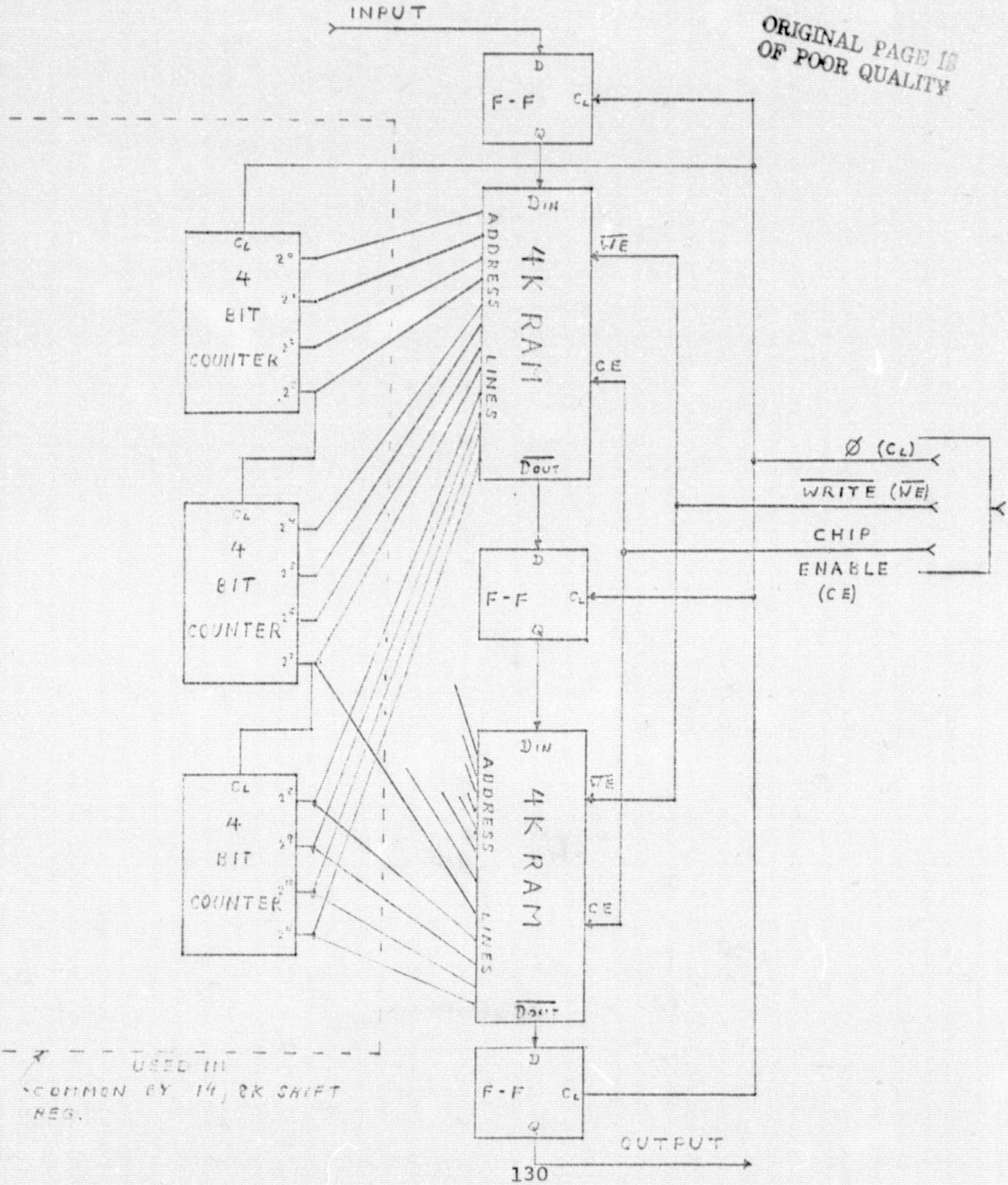
FIG 43 HIGH SPEED FRAME TO FRAME DELTA MODULATOR

were used as a shift register. Each memory IC has what is called a read-modify-write-cycle. In this mode an address is first sent to the memory (say address location one). The content of location one is read out of the memory and then the first bit of data is written into location one. Next the content of location two is read out of the memory and the second data bit is written into location two. This process would continue for N successive memory locations where N is the length of the shift register to be simulated by the memory. After storing N data bits and reading out N bits the memory returns to location one and reads out the first data bit. Then the N + 1 data bit is written into location one and the process is repeated again. Notice that the delay time between first storing a data bit and reading out the data bit is N read-modify-write-cyles of the memory. Thus the memory appears as a serial input serial output shift register, N bits long and with a clock time equal to one read-modify-write-cycle. Figure 44 shows 4K random access memories cascaded to produce one longer shift register.

Since the read-modify-write-cycle of these memories takes 1 us, the maximum shift rate of the shift register is only 1 MHz. To achieve the required 8 MHz shift rate, eight 24K shift registers were multiplexed as shown in Figure 45 to produce one 200K bit long 8 MHz shift register. Timing signals and synchronization were provided by the circuits shown in Figures 46 and 47. Pictures of the circuit boards are shown in Figure 48 and part of the assembled system is shown in Figure 49.

When the interframe delta modulator was used on real time motion

Fig. 44 One 8K Shift Register Made From 8K of Random Access Memory



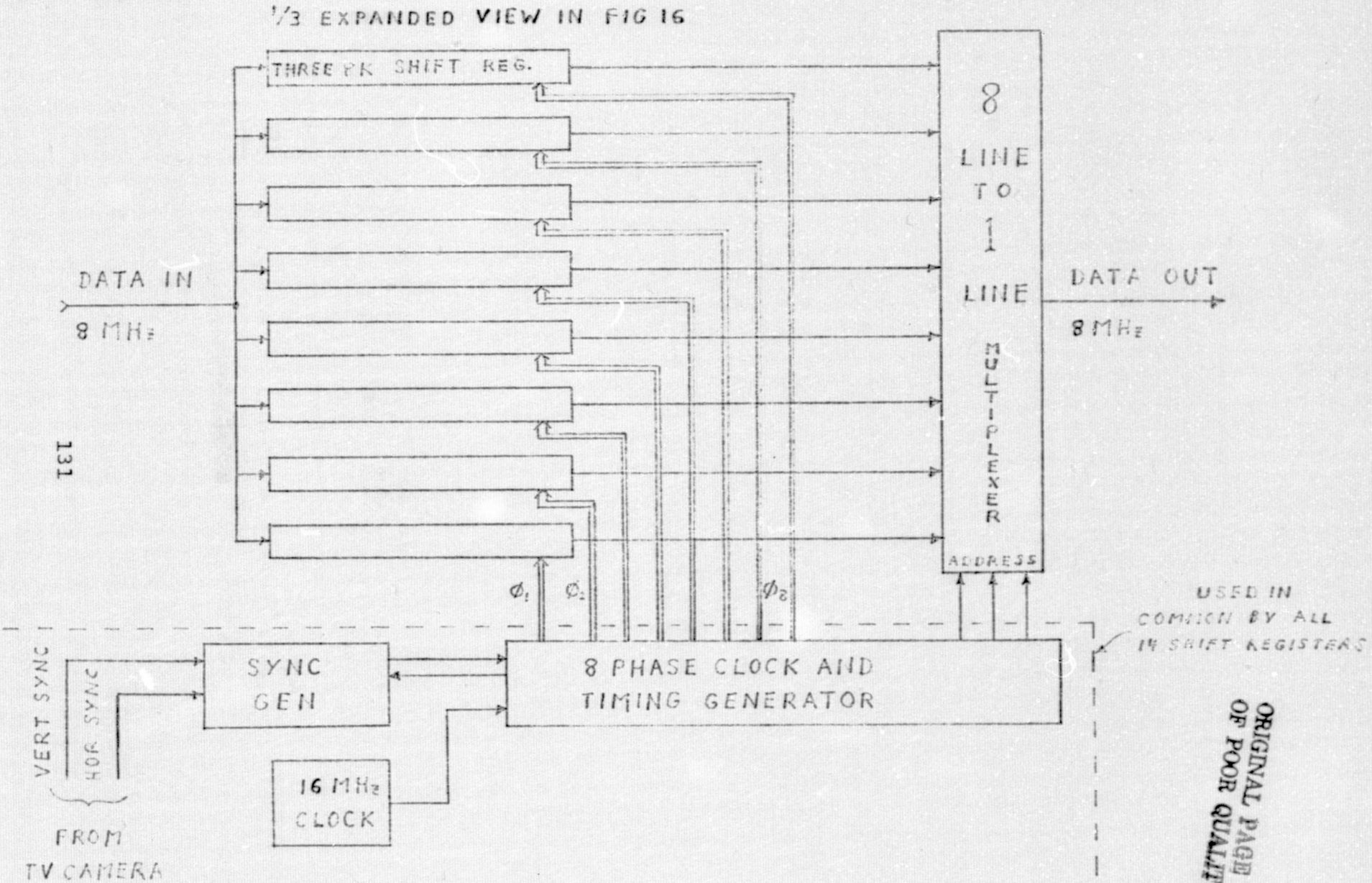


Fig. 45 One of Fourteen, 250,000 Bit Long Shift Registers

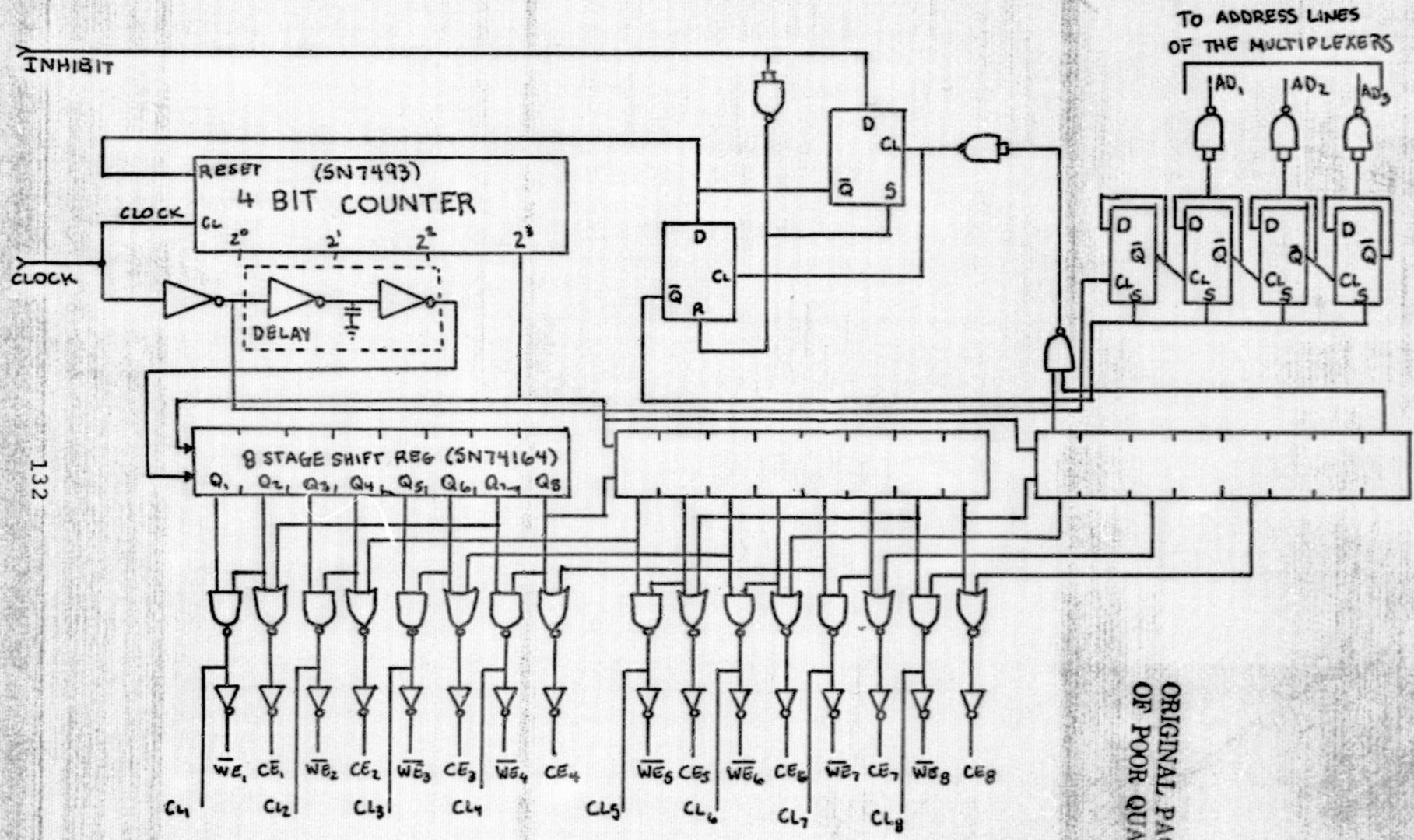


FIG 46 8STAGE CLOCK AND TIMING GENERATOR

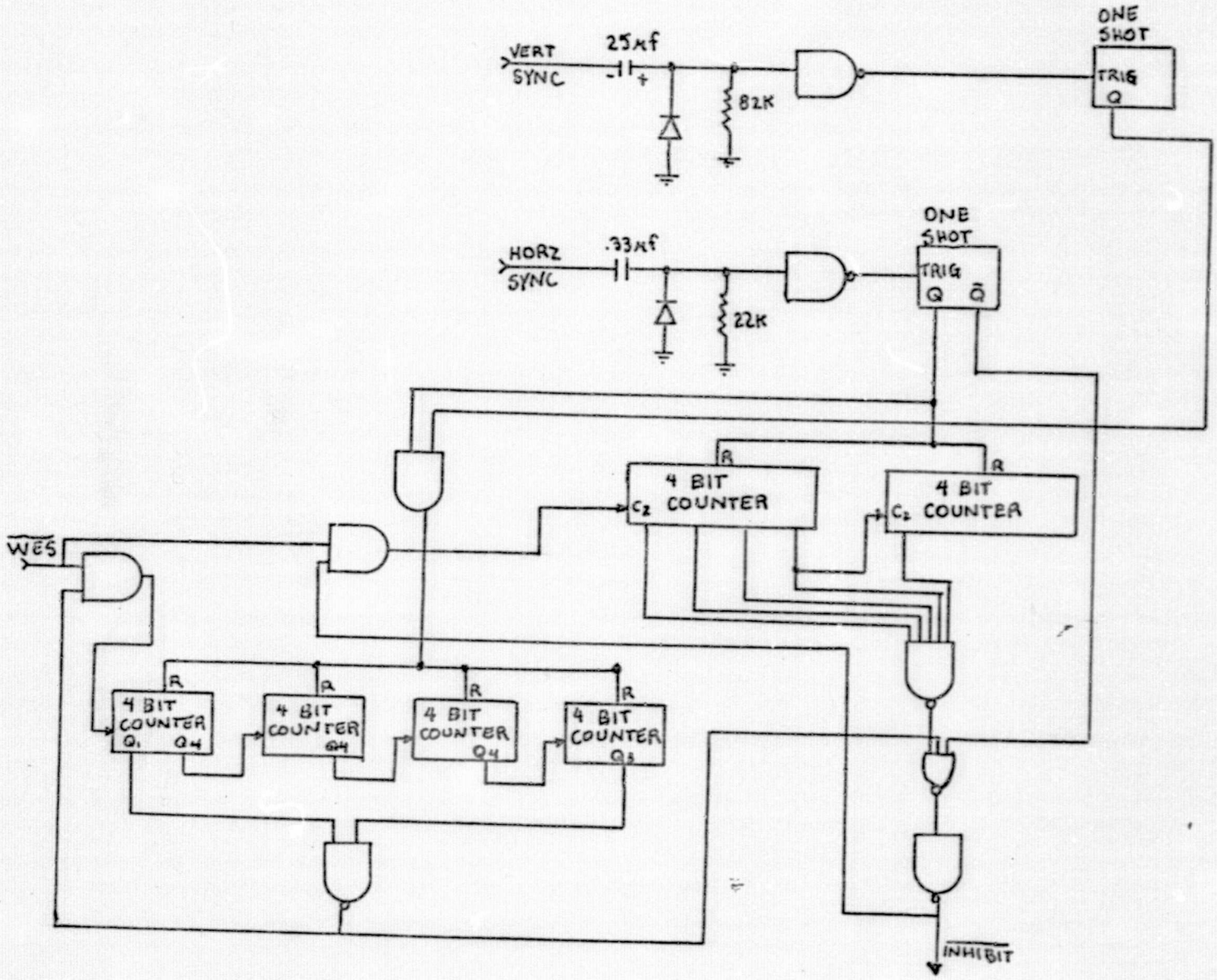
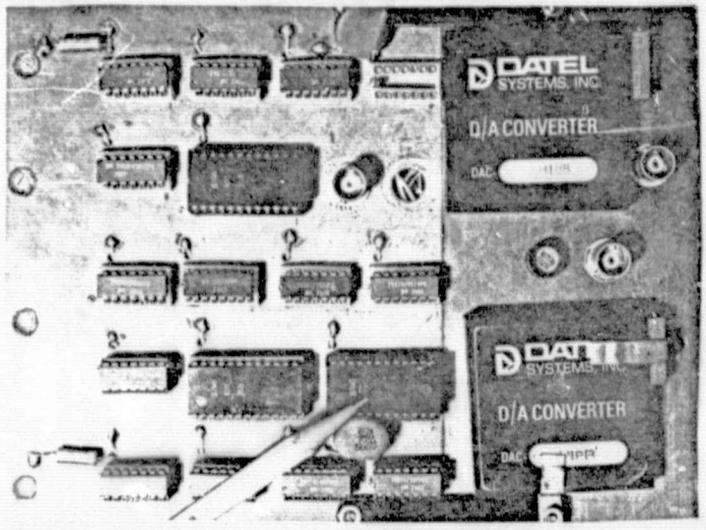
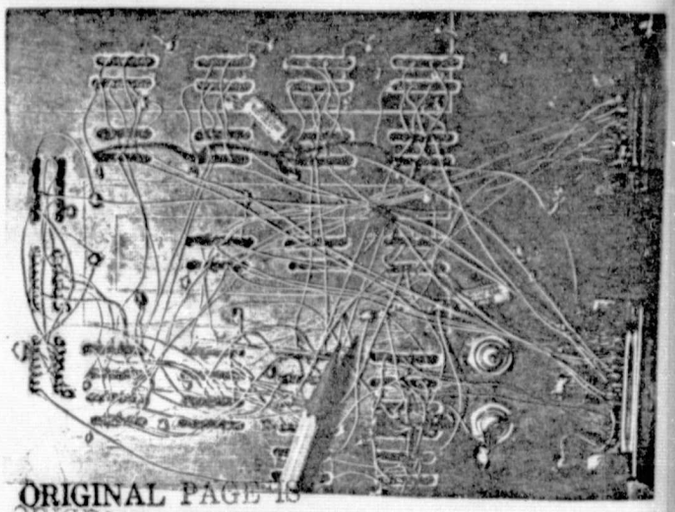


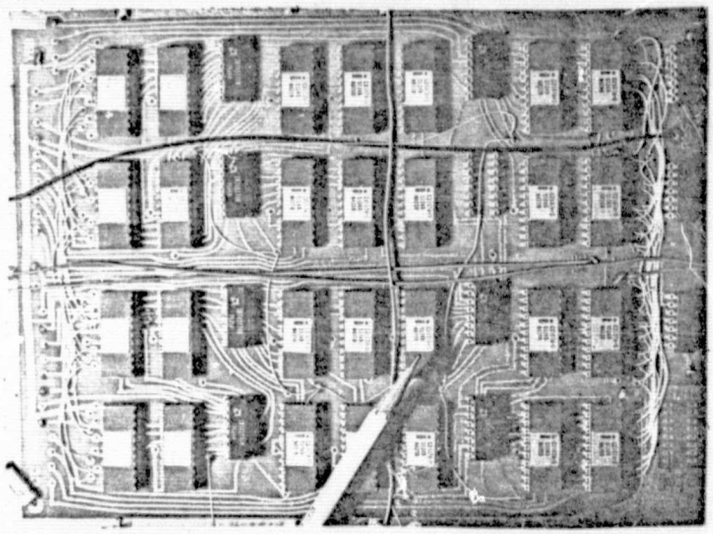
FIG 47 SYNC. GENERATOR

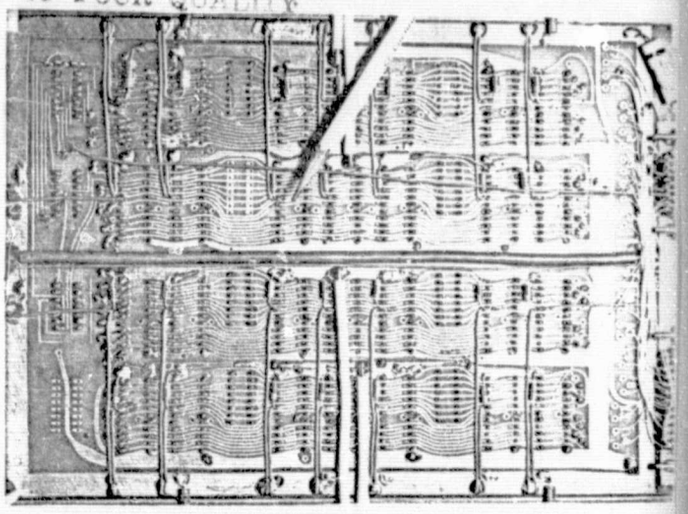
Fig. 48 Circuit boards for the frame to frame delta modulator
Clockwise from top left: Delta modulator; Timing generator;
One of 16 shift register boards - bottom view; Sync generator;
Multiplexer; Shift register board-top view.

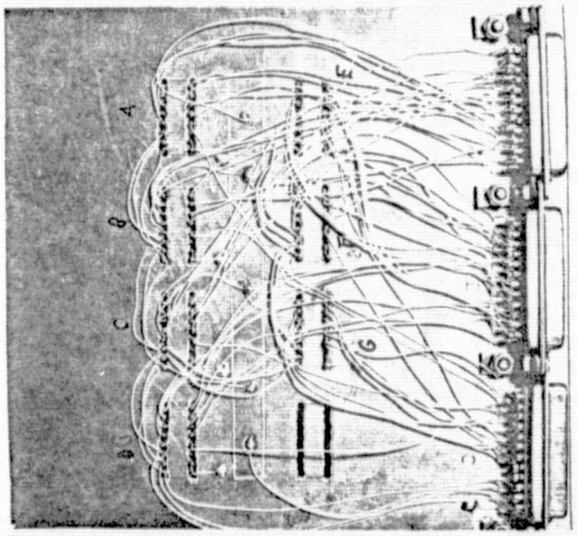




OF POOR QUALITY







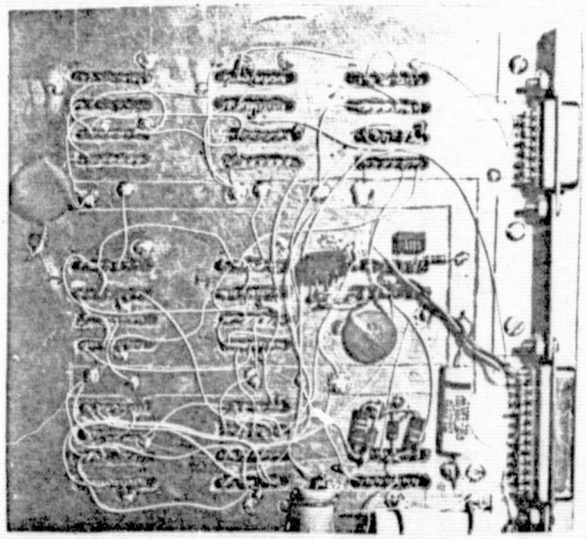
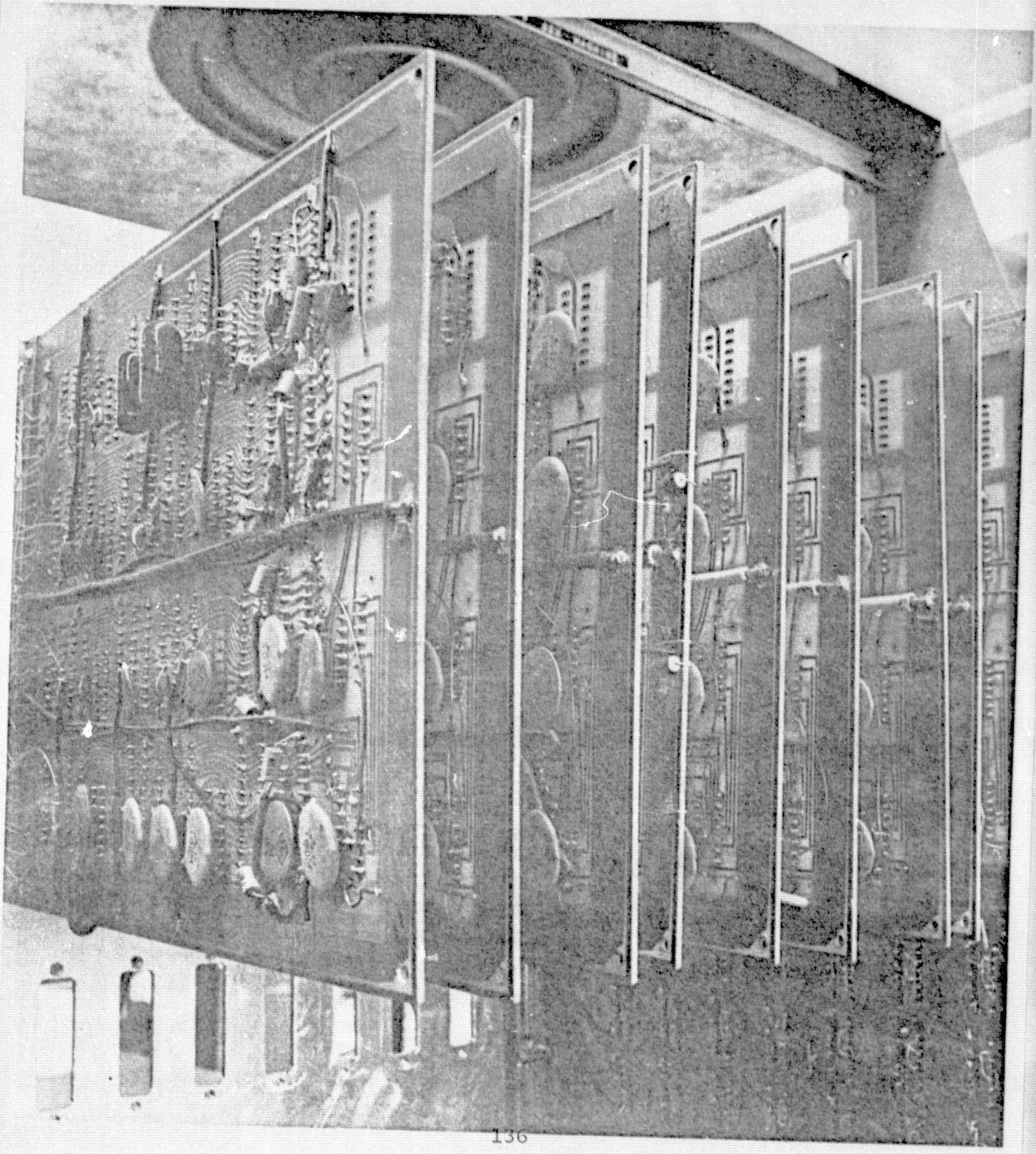
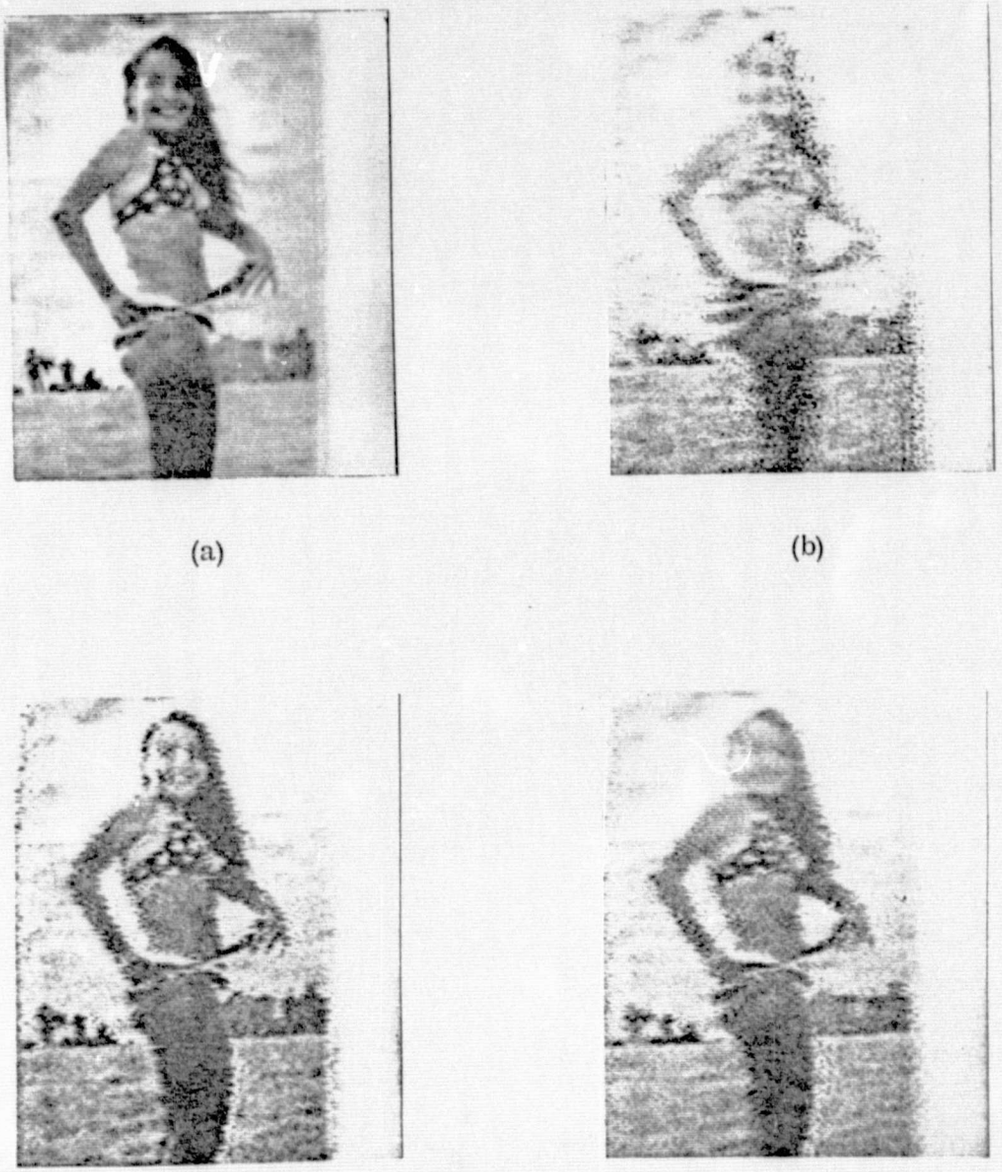


Fig. 48

Fig. 49 Interframe Delta Modulator Assembly





(a) (b)

(c) (d)

Fig. 50 Interframe Encoder (a) No motion (b) Motion
One dimentional Encoder (c) No motion (d) Motion

picture TV signals the encoded pictures were as good as the original picture when there was no motion. In those parts of the picture where motion occurs the picture severely degrades. It requires about 1/3 of a second for the picture to recover from movement. Figure 50a shows a still picture encoded with the interframe delta modulator. In Figure 50b we can see how the picture breaks up when the camera pans a scene. For comparison the regular delta modulator described in section 5.4 is shown in Figure 50c encoding a still picture at the same sampling rate as the interframe delta modulator. In Figure 50d the regular delta modulator is shown encoding a motion picture under the same conditions as in Figure 50b. Figure 50 shows that the interframe delta modulator out performs the regular delta modulator on still scenes but performs much worse on moving scenes.

It is our opinion that the interframe delta modulator in the form described in this section will not produce satisfactory pictures in most applications. An exception to this would be the display of a black board or other graphic material where there was no motion for intervals of several seconds. In section 6.3 a modification of the interframe delta modulator will be described which will enable the interframe delta modulator to produce satisfactory pictures.

6.2 Two Dimensional Delta Modulation

By taking advantage of the fact that pictures are two dimensional

arrays with each pel correlated to all the pels around it, it is possible to design a delta modulator which does not exhibit edge business. In Figure 14 it is shown that vertical edges exhibit edge business but horizontal edges do not. Vertical edges exhibit edge business because they are at right angles to the direction that the delta modulator encodes whereas horizontal edges lie in the direction that the delta modulator encodes. (A delta modulator normally encodes a picture from left to right because TV cameras scan from left to right.) Hence, vertical edges present themselves to the delta modulator as a step like change in the video signal whereas horizontal edges appear as a DC level. If a delta modulator could be designed to encode horizontally on horizontal edges and encode vertically on vertical edges then edge business on horizontal and vertical edges could be eliminated.

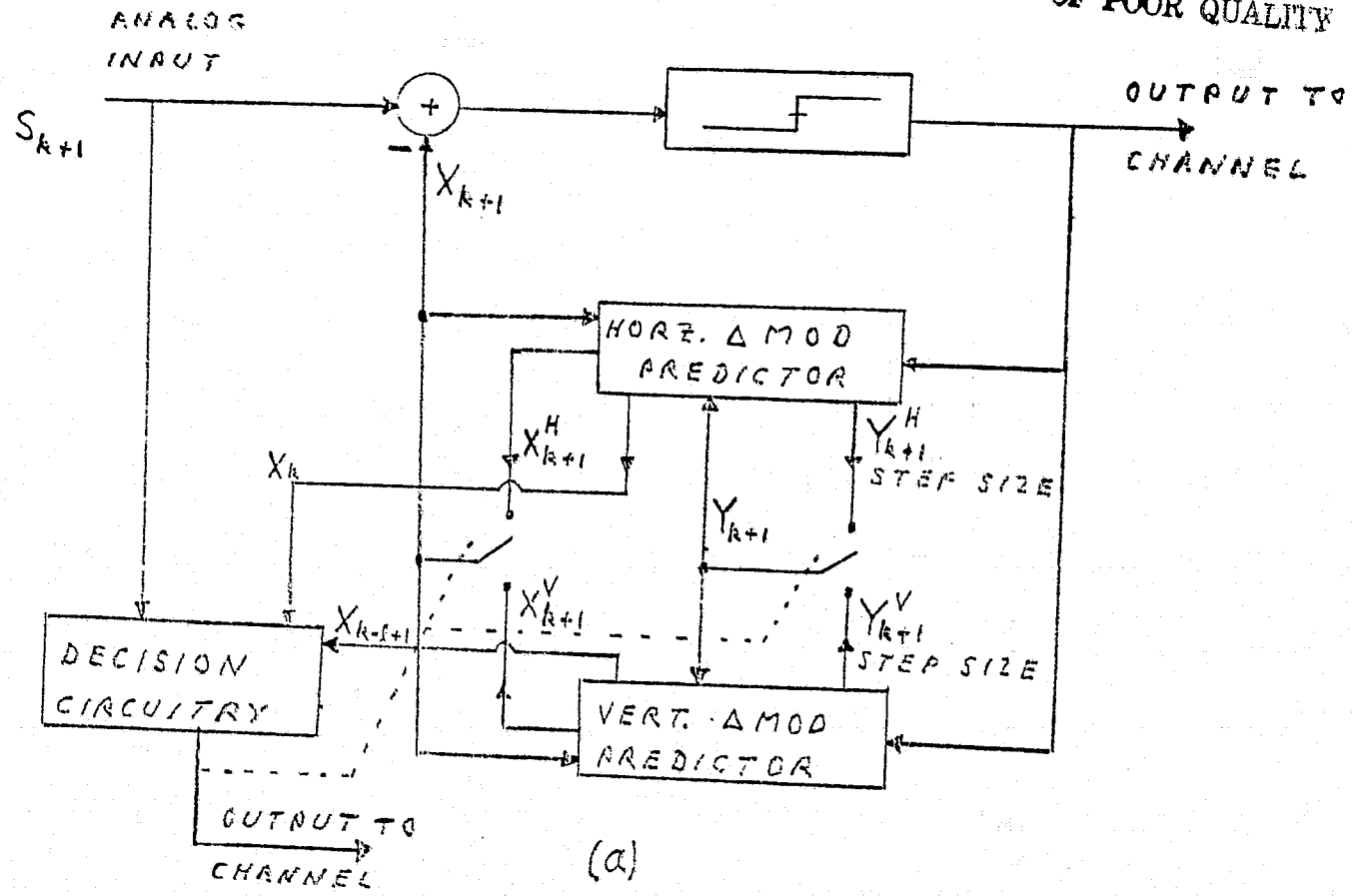
Another way to express this idea would be to say that a pel along a vertical edge is uncorrelated to the pels to its right but is correlated to the pels directly below. Likewise, a pel along a horizontal edge is uncorrelated to the pels below but is correlated to the pels directly to the right. If a delta modulator was designed to follow an encoding path of maximum correlation between pels then edge business would be minimized.

Such a delta modulator was designed and built to encode real time TV pictures. It was called a "two dimensional delta modulator" since it encodes from both the horizontal or vertical direction. (The delta modulator described in section 5.4 is termed a one dimensional delta

modulator since it encodes only in the horizontal direction.)

The operation of the two dimensional delta modulator can be understood by referring to Figure 51. Figure 51a shows that the two dimensional delta modulator contains a horizontal predictor and a vertical predictor. It also contains decision circuitry which decides whether the horizontal predictor or the vertical predictor is to be used to form the next delta modulator's output (estimate, X_{k+1}). Figure 51b shows the location in a picture of the estimate, step size, and E_k stored in the horizontal and vertical predictors. X_{k+1} is the current sample to be formed by the delta modulator. X_k, Y_k, E_k are the estimate, step size, and E_k in the horizontal predictor and $X_{k-\ell+1}$, $Y_{k-\ell+1}$ are stored in the vertical predictor. Using X_k, Y_k and E_k the horizontal predictor forms X^H_{k+1}. Likewise the vertical predictor forms X^V_{k+1} from $X_{k-\ell+1}$, $Y_{k-\ell+1}$ and $E_{k-\ell+1}$. The predictors form X^H_{k+1} and X^V_{k+1} according to equations 3.1.1, 3.1.2 and 3.1.3 which are the equations that described the one dimensional delta modulator. Thus the horizontal predictor is the same as the one dimensional delta modulator's predictor, and the vertical predictor is also the same as the one dimensional delta modulator's predictor except for the fact that one scanning line of storage is required to store $X_{k-\ell+1}$, $Y_{k-\ell+1}$ and $E_{k-\ell+1}$ because these values were formed during the previous scanning line.

The decision circuitry decides whether the delta modulator will let X_{k+1} and Y_{k+1} be equal to X^H_{k+1} and Y^H_{k+1} or X^V_{k+1} and Y^V_{k+1}. If X^H_{k+1} and Y^H_{k+1} are chosen to be X_{k+1} and Y_{k+1} then X^H_{k+1} and Y^H_{k+1} are stored



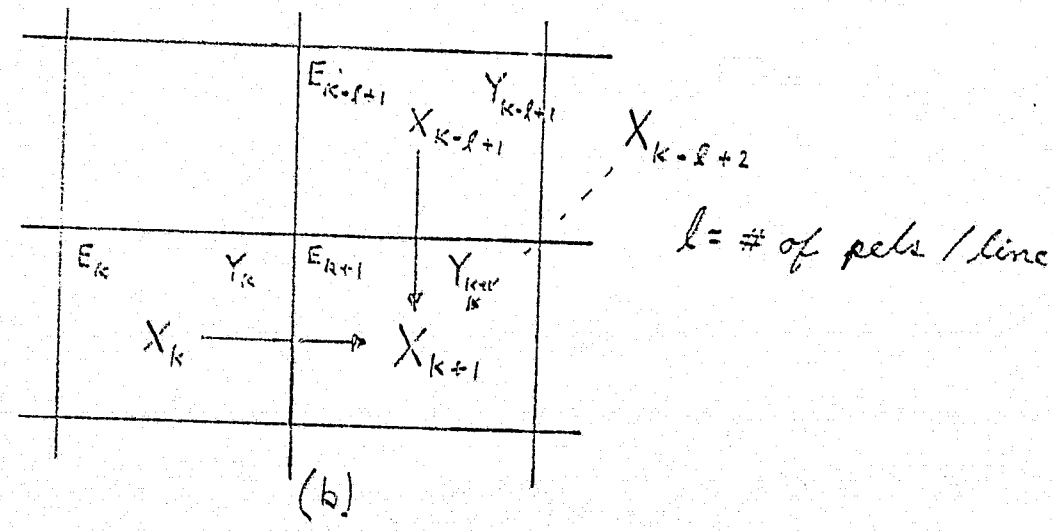


FIG 5| TWO DIMENSIONAL Δ MOD

in both the horizontal and vertical predictor for the next prediction, and X^V_{k+1} and Y^V_{k+1} are discarded. If X^V_{k+1} and Y^V_{k+1} were chosen then the reverse would be true. The decision circuitry makes its decision by comparing X_k and $X_{k-\ell+1}$ with the input signal S_{k+1}. If S_{k+1} is closer in amplitude to X_k than to $X_{k-\ell+1}$, then the delta modulator will encode horizontally, forming X_{k+1} from X^H_{k+1}. If S_{k+1} was closer to $X_{k-\ell+1}$ then the delta modulator would encode vertically and form X_{k+1} from X^V_{k+1}. An extra bit of information must be sent to the receiver to inform the receiver as to which decision the decision logic makes. Thus the two dimensional delta modulator transmits two bits for each sample of the video signal.

The set of equations describing the two dimensional delta modulator are given below:

Let D^H_k and D^V_K be defined as follows:

$$D^H_k = |S_{k+1} - X_k| \quad \text{and} \quad D^V_k = |S_{k+1} - X_{k-\ell+1}|$$

Then

$$E_{k+1} = \text{Sgn}\left[S_{k+1} - X_k\right] \; ; \; D^H_k < D^V_k$$

$$E_{k+1} = \text{Sgn}\left[S_{k+1} - X_{k-\ell+1}\right] \; ; \; D^H_k > D^V_k$$

$$X_{k+1} = X_k + Y^H_{k+1} \quad ; \quad D^H_k < D^V_k$$

$$X_{k+1} = X_{k-\ell+1} + Y^V_{k+1} \quad ; \quad D^H_k > D^V_k$$

$$Y^H_{k+1} = \begin{cases} |Y_k|(E_{k+1} + \tfrac{1}{2}E_k) & ; \quad |Y_k| \geq 2 \\ 2E_{k+1} & ; \quad |Y_k| < 2 \end{cases}$$

$$Y^V_{k+1} = \begin{cases} |Y_{k-\ell+1}|(E_{k+1} + \tfrac{1}{2}E_{k-\ell+1}) & ; \quad |Y_{k-\ell+1}| \geq 2 \\ 2E_{k+1} & ; \quad |Y_{k-\ell+1}| < 2 \end{cases}$$

The implementation of the decision circuitry of Figure 51 posed a special problem because the decision must be made within 25ns if the delta modulator is to work in real time. The solution to this problem is shown in Figure 52. The output of the exclusive or gate is zero if the input signal is closer to X^H, and one if the input signal is closer to X^V.

The circuit schematic of the delta modulator is shown in Figure 53. The schematic does not include the one scanning line of memory used to store $X_{k-\ell+1}$, $Y_{k-\ell+1}$ and $E_{k-\ell+1}$ nor does it include the phase lock loop required to sync the delta modulator's 8 MHz clock to the horizontal sync signal. The total number of IC's used to implement the two dimensional delta modulator algorithm was about 50 IC's. This makes the

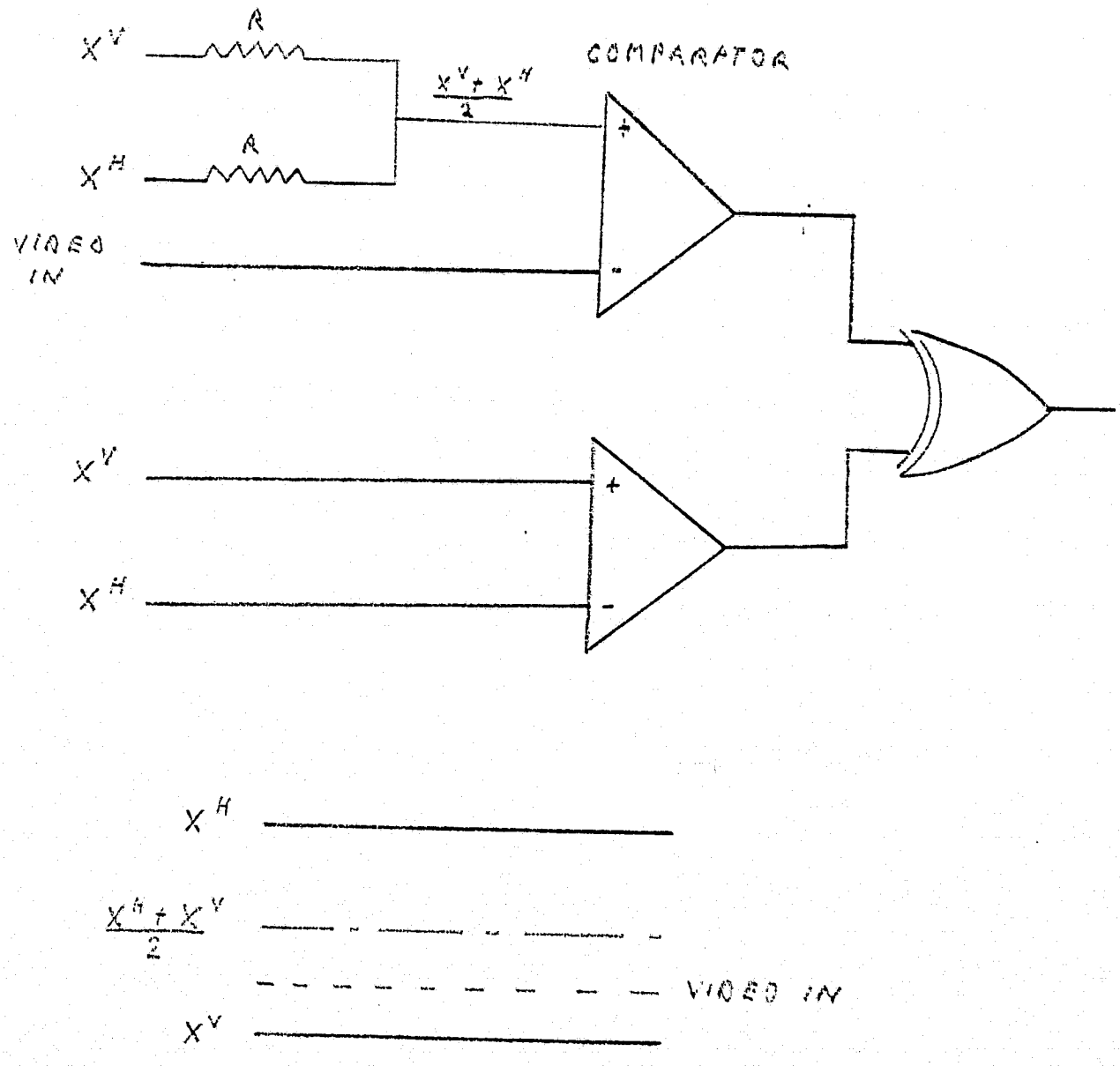


FIG 52

DECISION CIRCUITRY

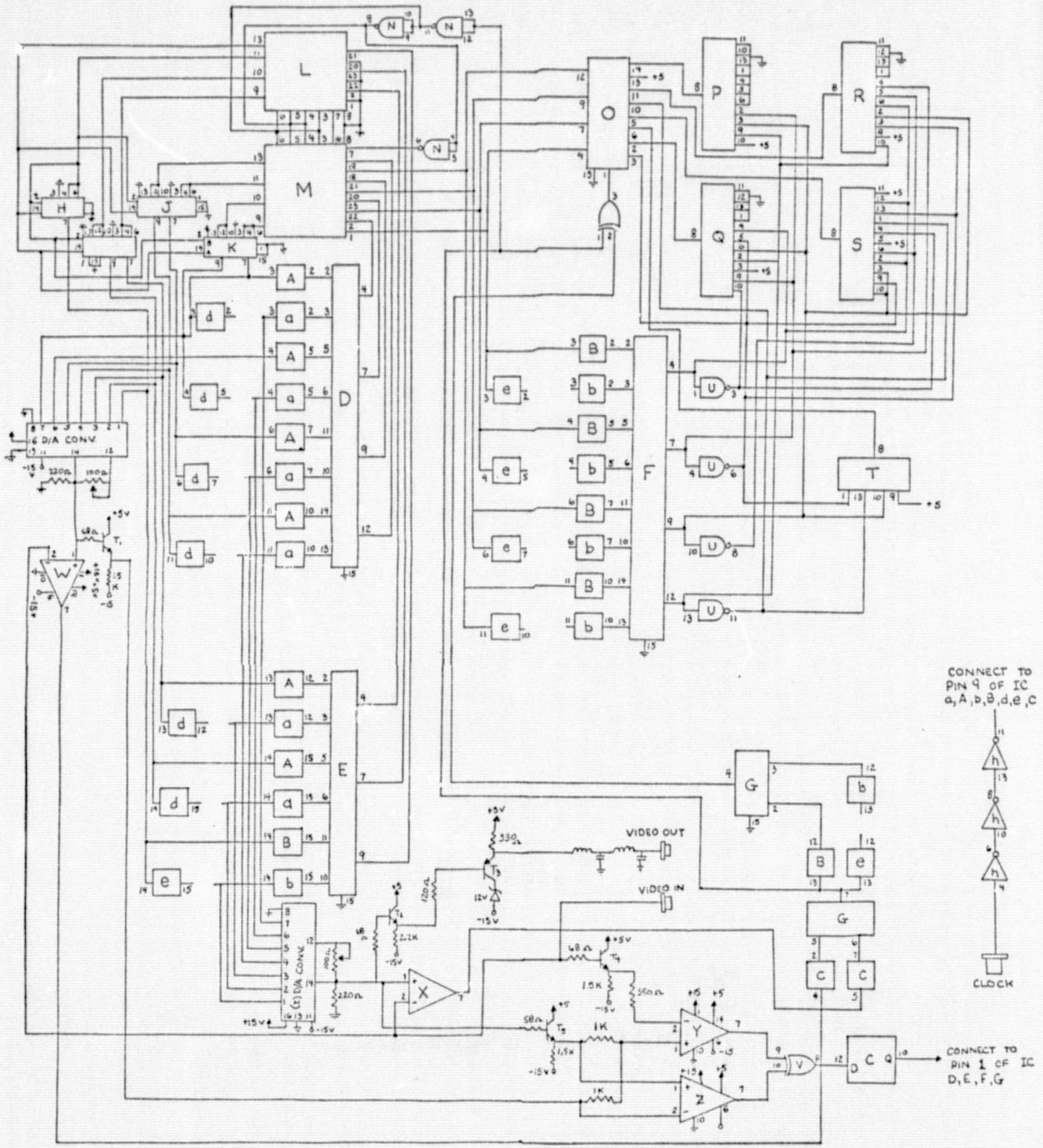


FIG 53 TWO DIMENSIONAL DELTA MODULATOR

two dimensional delta modulator 2.5 times as complex as the one dimensional delta modulator.

When the delta modulator was tested on real time motion pictures from an NTSC standard black and white TV camera the delta modulator degraded in an unexpected way. Apparently each frame of the motion picture (especially at the corners of objects and along diagonal lines with positive slope) was encoded slightly differently. This caused flickering at the corners of objects and along diagonal lines with positive slope. This phenomenon is caused by the fact that at a corner neither X_k nor $X_{k-\ell+1}$ is a good estimate of S_{k+1}, the corner pel. Therefore, several pels must go by before the delta modulator acquired the value of the corner pel. An analogous situation exists for diagonal lines with positive slope. In effect, the delta modulator slope overloads on these signals. The slope overloading of the two dimensional delta modulator is worse than for the one dimensional delta modulator because the two dimensional delta modulator has only one half the number of samples per pel to achieve the same transmission bit rate as the one dimensional delta modulator.

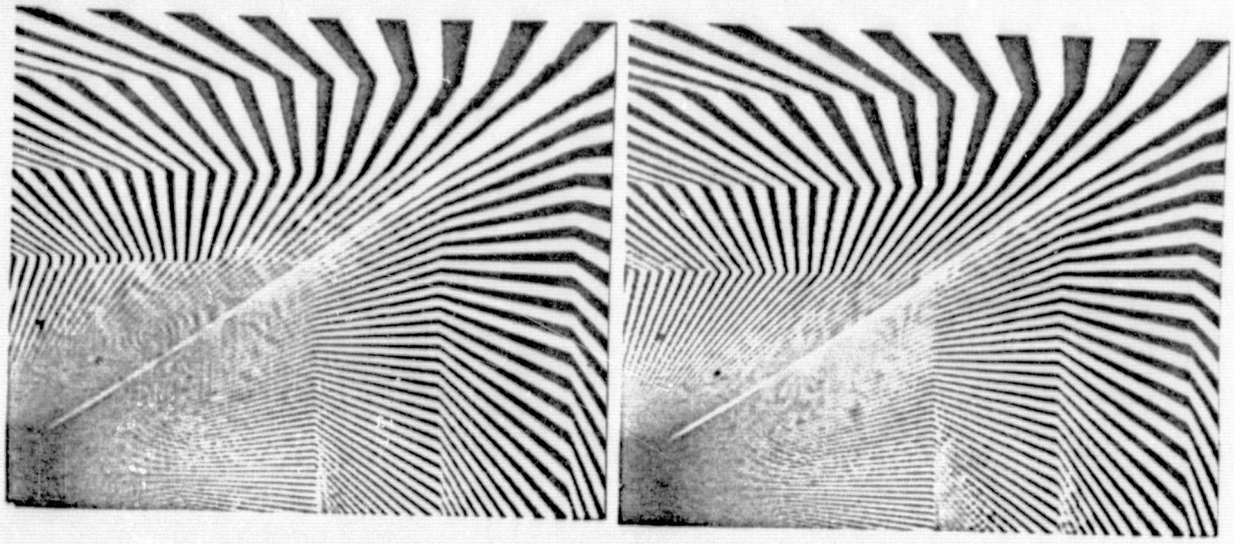
To improve the performance of the delta modulator on diagonal lines with positive slope the delta modulator was modified so that instead of encoding horizontally or vertically with X_k or $X_{k-\ell+1}$, the delta modulator could encode horizontally or diagonally choosing X_k or $X_{k-\ell+2}$. We termed this new encoding scheme advanced encoding since $X_{k-\ell+2}$ is one pel ahead of $X_{k-\ell+1}$ (see Figure 51b). A comparison of the two

encoding schemes is shown in Figures 54a and b. Notice that in Figure 54a horizontal and vertical lines are encoded perfectly but diagonal lines with positive slopes of around 45° are jagged. In Figure 54b with advanced encoding horizontal edges are perfect, vertical edges are almost perfect and diagonal lines are substantially improved. Over all, advanced encoding produces better pictures than non-advanced encoding.

Subjective comparisons of the one and two dimensional delta modulators at 2 bits per pel (16 MHz transmission rate) has shown them to produce pictures of nearly identical quality. Since the one dimensional delta modulator is less complex than the two dimensional delta modulator this author feels that for most applications the one dimensional delta modulator would be preferred. An exception to this would be the transmission of a single frame because the flickering caused by the fact that each frame is encoded differently would be absent.

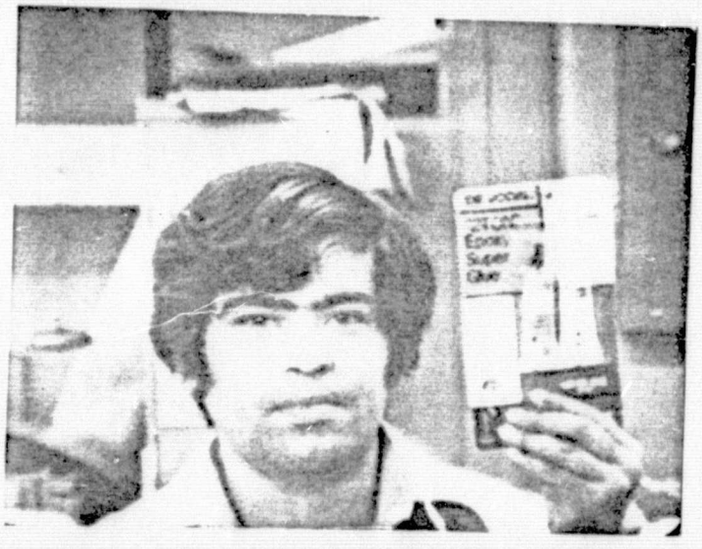
6.3 Interframe Two Dimensional Delta Modulation

The principle of two dimensional delta modulation can be applied to the interframe encoder. The interframe encoder was modified so that the current estimate could be formed from either the estimate exactly one frame before, or one sample before. If the estimate in the previous frame is chosen to form the current estimate then the encoding procedure is the same as the interframe encoding of section 6.1. If, however,



(a) Nonadvanced encoding

(b) Advanced encoding



(c) Advanced encoding

Fig 54 Two dimensional delta modulator transmitting at 18mb/s with non-advanced and advanced encoding.

the estimate on the same frame, one sample to the left is chosen to form the current estimate, then the encoding procedure is the same as the one dimensional delta modulator. As with the two dimensional delta modulator of section 6.2, the choice as to which estimate is to be used to form the current estimate is determined by which estimate is closer in value to the current pel. Two bits are sent to the receiver for each sample of the video signal. The first bit is the delta modulator's output, E_k, and the second bit informs the receiver as to which estimate the transmitter chose to form the current estimate.

Figure 55 shows a block diagram of the encoder. The block diagram of the two dimensional interframe encoder of Figure 55 is functionally the same as the two dimensional encoder of Figure 51 except that the vertical predictor with one line of memory has been replaced with a frame to frame predictor with one frame of memory.

The result of using the interframe two dimensional delta modulator on real time TV signals with a transmission rate of 16 Mb/S is shown in Figure 56. Figure 56a shows a resolution chart with no motion. There is no degradation. Figure 56d shows the effects of panning the scene. When the camera pans the scene there is so much change in the pels between frames that the delta modulator almost always chooses the estimate to right within the same frame to encode the current estimate. Therefore, on movement the delta modulator tends to encode as if it were a one dimensional delta modulator operating at 8 Mb/S. When there is no movement the delta modulator tends to encode as the interframe

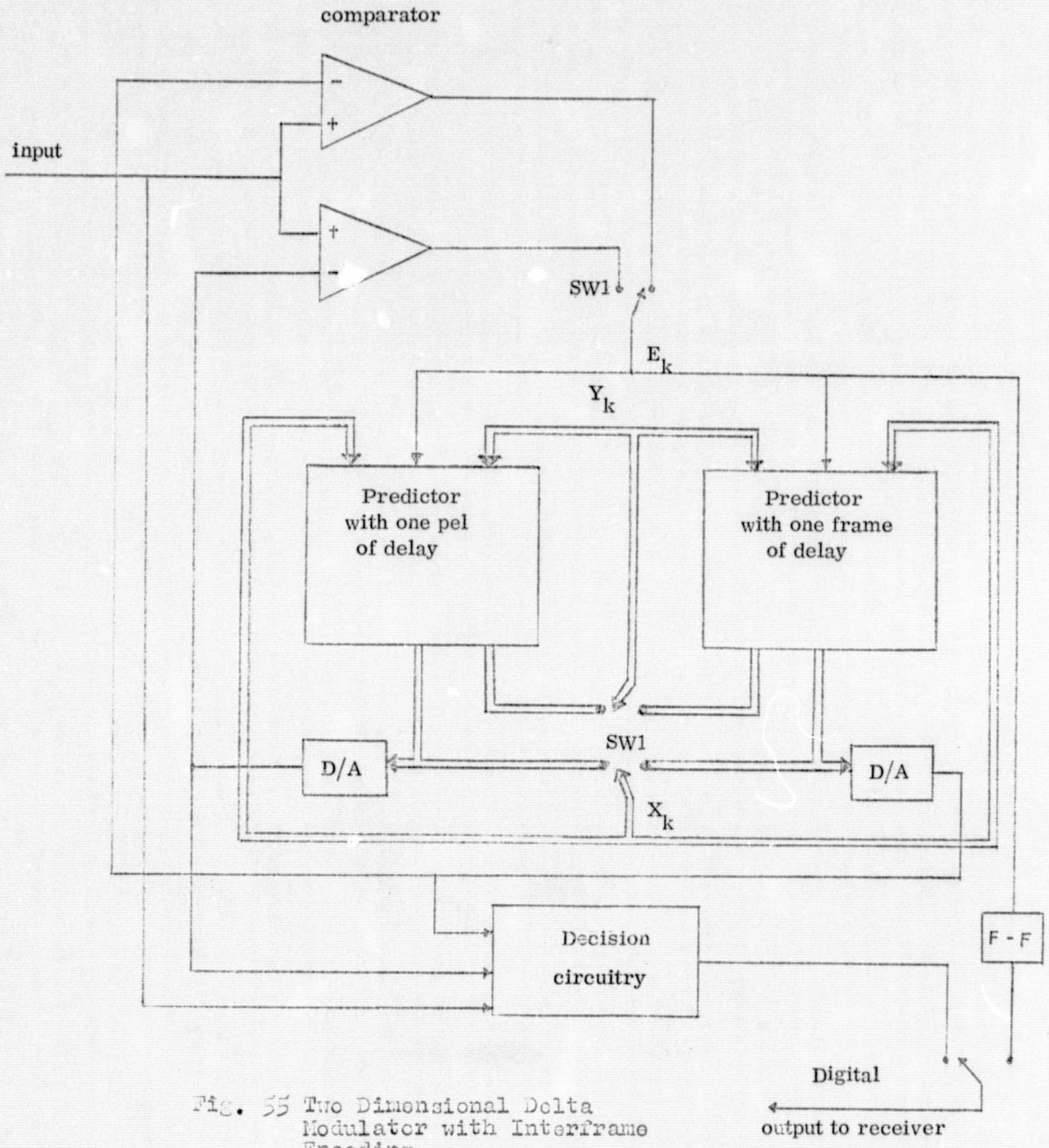
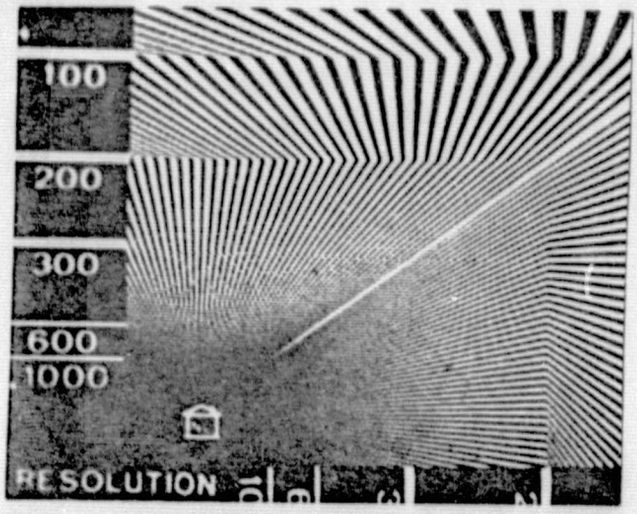


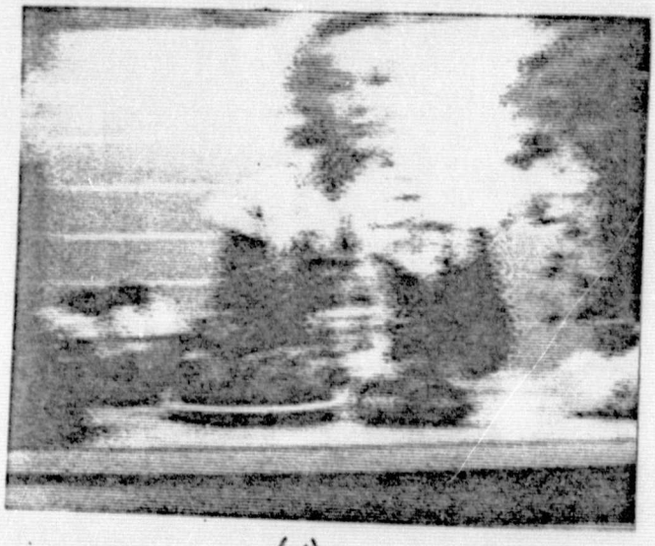
Fig. 55 Two Dimensional Delta Modulator with Interframe Encoding



(a)

(b)

(c)



(d)

(e)

Fig. 56 Two Dimensional Interframe Encoder (a) Resolution Chart, no motion; (b) Girl, no motion; (c) Original analog video; (d) Effects of "panning" the scene (e) Analog video with "panning.

delta modulator of section 6.1.

For scenes with a moderate amount of motion the two dimensional interframe delta modulator operating at 16 Mb/S produces better quality pictures than any of the other delta modulation techniques. This delta modulator is much more complex than the other delta modulators. It required over 600 IC's to implement.

APPENDIX A

The factor of 7 db was calculated under the following conditions and constraints. It is well known from the psychophysics of vision that satisfactory pictures can be made with only 64 different brightness levels. Thus the number of bits (N) per PCM word required is $\log_2 64$ or

$$N = 6 \text{ bits/sample} \tag{A.1}$$

The probability of the occurrence of a bit error (Pe) due to white gaussian (1) noise is

$$Pe = \tfrac{1}{2}\text{erfc}\ \frac{Es}{n_o} \tag{A.2}$$

where Es is the energy per bit and n_o is the power spectral density of the noise at the receiver's input. A typical Pe for a digital communications channel is $Pe = 10^{-5}$ which, from Eq (1.2), yields

$$\frac{Es}{n_o} = 9 \text{ db} \tag{A.3}$$

Es can be related to the received power, S_{pcm}, by [1]

$$Es = \frac{S_{pcm}}{2f_m N} \tag{A.4}$$

where fm is the highest frequency component in the baseband TV signals. Combining Eq A.3 and Eq A.4 yields the value of signal-to-noise ratio at the receiver's input required to achieve a probability of error of 10^{-5}.

$$\frac{S_{pcm}}{n_o f_m} = 20 \text{ db} \tag{A.5}$$

The signal-to-noise ratio at the output of the receiver (S_o/N_o) with Pe = 10^{-5} is given by: (Taub and Schilling, Prin. of Comm. Sys.)

$$\frac{S_o}{N_o} = \frac{2^{2n}}{1+4Pe2^{2n}} = 35 \text{ db} \tag{A.6}$$

Eq A.6 includes both the effects of bit errors and quantization noise.

We will now compute the ratio of transmitted power (S_{FM}) in the FM system, to that of PCM (S_{pcm}), which will yield the same signal-to-noise ratio (S_o/N_o) at the output of both systems. The comparison will be made with the bandwidth of the FM system the same as that of the PCM system. The bandwidth (BW) of the PCM system and, hence, also the FM system is given by

$$B.W. = f_m N = 6 f_m \tag{A.7}$$

From Carson's rule and Eq 1.7 we can find the modulation index, B, of the FM signal

$$B.W. = 6\ f_m = 2(B+1)f_m$$

$$B = 2 \qquad (A.8)$$

Substituting from Eq A.6 and A.8 we can solve Eq A.9 for the signal-to-noise ratio ($S_{FM}/n_o fm$) at the input to the FM receiver such that the output signal-to-noise ratio (S_o/N_o) is the same in both FM and PCM

$$\frac{S_o}{N_o} = \frac{3}{2} B^2 \frac{S_{FM}}{n_o f_m} \qquad (A.9)$$

$$\frac{S_{FM}}{n_o f_m} = 3162\left(\frac{2}{3}\right)\left(\frac{1}{2}\right)^2 = 27 \text{ db} \qquad (A.10)$$

Subtracting Eq A.5 from A.10 shows that the PCM system can achieve the same signal-to-noise ratio as the FM system with 7 db less power.

REFERENCES

1. Wintz, "Transform Picture Coding", Proceeding of the IEEE, vol. 60, no. 7, July 1972.

2. Schreiber, "Picture Coding", Proceeding of the IEEE, vol. 55, no. 3, March 1967.

3. Seyler, "The Coding of Visual Signals to Reduce Channel Capacity", IEEE Monograph #535E, July 1962.

4. Pratt, "Slant Transform Image Coding", IEEE Transactions on Communications, August 1974.

5. Linkibit, "Digital TV Processing System Final Report", Nov. 26, 1975.

6. O'Neal, Jr., "Predictive Quantizing Systems for the Transmission of TV Signals", The Bell System Tech. Journal, MAY-JUNE, 1966.

7. R. Lippmann, "A Technique for Channel Error Correction in Differential PCM Picture Transmission", IEEE Int. Conf. Commun., Conf. Rec. vol.11, June 11-13, 1973.

8. T. Ishiguro, "Interframe Coding for 4MHz Color Television Signals", Conference Record ICC, June 16-18, 1975.

9. J.C. Candy, "Transmitting Television as Clusters of Frame-to-Frame Differences", BSTJ, vol. 50, no. 6, July-August, 1971.

10. Song, Garodnick, and Schilling, "A Variable Step Size Robust Delta Modulator", IEEE Trans. Commun., vol. COM-19, pp. 1033-1044.

II. DOCTORAL STUDENTS GRADUATED

Tuvia Apelewicz, 1978, Thesis title, "The Design of a Programmable Real Time ADM Voice Processor and the Formulation of the Autocorrelation Function for the LDM Algorithm".

Richard Lei, 1978, Thesis title, "Delta Modulation Encoding of Video Signals".

Norman Scheinberg, 1979, Thesis title, "The Delta Modulation of Video Signals".

III. PAPERS PUBLISHED

1. Voice Encoding for the Space Shuttle Using Adaptive Delta Modulation, D. L. Schilling, J. Garodnick, and H. A. Vang, IEEE Transactions on Communications, November 1978, No. 11, Vol. COM-26.

2. Video Encoding Using Adaptive Delta Modulation, D. L. Schilling, N. Scheinberg and J. Garodnick, IEEE Transactions on Communications, November 1978, No. 11, Vol. COM-26.